Penerapan Matriks Rotasi dalam Menganalisis Pergerakan Karakter dalam Gim 2D

Muhammad Hazim Ramadhan Prajoda - 13523009
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
hazimmuhammad28@gmail.com, *13523009@std.stei.itb.ac.id*

***Abstrak*—Pada pengembangan gim 2D, transformasi linier matriks rotasi mempunyai peran yang penting untuk membuat pergerakan karakter yang halus dan realistis. Makalah ini membahas tentang penerapan matriks rotasi dalam menganalisis pergerakan karakter dalam gim 2D dengan studi kasus gim Holocure. Implementasi percobaan menggunakan bahasa python dan library pygame dengan fitur rotasi tiga mode: rotasi mengikuti target, rotasi di tempat, dan rotasi orbit. Hasil dari implementasi ini mendemonstrasikan keefektifan dari matriks rotasi dalam membuat pergerakan karakter yang akurat dan efisien pada gim 2D modern.**

***Kata Kunci*—gim 2D, matriks rotasi, pergerakan karakter, transformasi linier.**

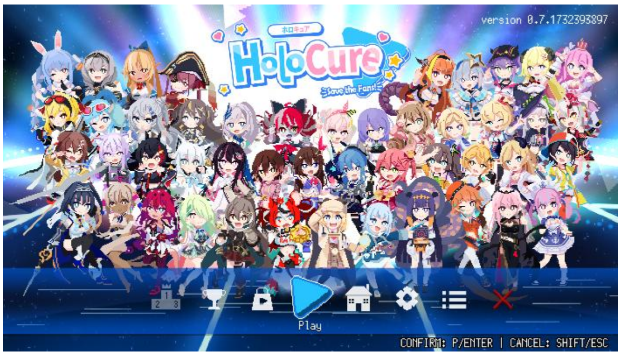
I. PENDAHULUAN

Pernahkah kamu bermain gim 2 dimensi (2D) dan bertanya-tanya bagaimana karakter kamu bisa berputar dengan mulus dan menghadap ke arah yang kamu tentukan? Sebenarnya, mekanisme ini bisa kita uraikan dengan penerapan konsep matematika, yaitu transformasi matriks atau lebih khususnya matriks rotasi. Matriks rotasi digunakan dalam memutar karakter atau objek dalam gim 2D tanpa mengubah ukuran atau mendistorsinya.

Gim 2D tetap diminati oleh berbagai kalangan karena menawarkan gaya visual yang khas dan alur permainan yang sering kali sederhana namun menantang. Dalam genre ini, kelancaran pergerakan karakter menjadi salah satu aspek penting yang mendukung pengalaman bermain yang optimal. Dalam gim 2D, pergerakan karakter dibatasi oleh dua dimensi pada sumbu koordinat kartesian-xy. Meskipun kelihatan sederhana, penerapan transformasi rotasi menjadi elemen yang penting untuk membuat pengalaman dalam bermain gim menjadi realistis dan menyenangkan. Dengan penerapan matriks rotasi, semua perputaran bisa dilakukan dengan presisi yang akurat membuat pergerakan yang mulus dan konsisten.

Selain itu, penerapan matriks rotasi juga mempengaruhi keputusan dan strategi pemain. Ketika gerakan objek berputar, pola yang dihasilkan akan berubah dibandingkan dengan gerakan objek yang lurus. Hal ini membuat pemain harus beradaptasi dengan pola gerakan tersebut untuk menghindarinya. Pemain tidak hanya harus mengendalikan karakter mereka, tetapi juga bisa merespons secara aktif terhadap lingkungan permainan yang terus berubah. Dengan memahami bagaimana objek berputar dan mempengaruhi pergerakan karakter, pemain dapat mengembangkan strategi yang lebih efektif untuk mencapai tujuan dalam permainan.

Fokus makalah ini akan menjelaskan secara matematis bagaimana penerapan matriks rotasi tersebut digunakan dalam gim 2D dengan memberikan salah satu contoh spesifik gim 2D yang ada yaitu Holocure. Lalu akan dianalisis pengaruh penerapan matriks rotasi ini dalam pergerakan karakter. Ini juga akan memberikan wawasan mengenai pentingnya transformasi matriks dalam gim modern.



Gambar 1. Gim Holocure, salah satu game dengan rating tertinggi di platform steam

II. DASAR TEORI

A. Teori Matriks

Definisi matriks

Matriks secara definisi adalah suatu *array* persegi panjang yang berisikan angka-angka. Matriks digunakan untuk mendeskripsikan suatu transformasi linier[1].

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Gambar 2.1. Matriks 2x2 secara umum

Setiap kolom dalam matriks merepresentasikan vektor basis hasil transformasi. Misal dalam Gambar 2.1, elemen a dan c merupakan komponen dari basis vektor pertama, sedangkan b dan d merupakan komponen basis vektor kedua. Matriks ini mendeskripsikan bagaimana ruang dua dimensi ditransformasikan melalui operasi linier.

Perkalian matriks

Perkalian suatu matriks bisa dibedakan menjadi 3, yaitu perkalian matriks dengan vektor, perkalian matriks dengan matriks lainnya, dan perkalian matriks dengan skalar.

Perkalian matriks dengan vektor adalah cara untuk menerapkan suatu transformasi linier (matriks) ke dalam vektor. Misal untuk matriks A dan vektor V sebagai berikut:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \qquad \vec{V} = \begin{bmatrix} u \\ v \end{bmatrix}$$

Maka hasil dari perkalian matriks dengan vektor adalah:

$$A\vec{V} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} au + bv \\ cu + dv \end{bmatrix}$$

Gambar 2.2. Perkalian Matriks dengan vektor

Perkalian matriks dengan matriks merepresentasikan penerapan beberapa transformasi linier secara berurutan. Hasil dari operasi ini adalah sebuah matriks yang merepresentasikan komposisi dari setiap transformasi yang dilakukan[2]. Misal untuk matriks A dan B, dengan elemen matriks B:

$B = \begin{bmatrix} e & f \\ g & h \end{bmatrix}$, hasil perkalian matriks A dengan B adalah:

$$AB = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix}$$

Gambar 2.3. Perkalian Matriks dengan Matriks

Matriks memiliki sifat tidak komutatif, artinya urutan perkalian matriks memengaruhi hasil yang didapat. Misal ada 2 matriks, A dan B, yang berukuran sama. Sifat tidak komutatif pada matriks akan mengakibatkan

$$AB \neq BA$$

Perkalian matriks dengan skalar akan menghasilkan matriks yang setiap elemen pada matriks tersebut merupakan perkalian dengan bilangan skalar. Misal bilangan skalar k dikalikan dengan matriks A

$$kA = k \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} ka & kb \\ kc & kd \end{bmatrix}$$

Gambar 2.4. Perkalian Matriks dengan skalar

B. Matriks Rotasi

Definisi matriks rotasi 2×2

Matriks rotasi adalah transformasi linier yang memutar arah/sudut suatu vektor tanpa mengubah ukuran/besar vektor tersebut. Matriks ini dapat diturunkan dengan mempertimbangkan sifat geometris rotasi. Untuk ruang 2D, matriks rotasi dengan perputaran sudut sebanyak θ, direpresentasikan sebagai berikut:

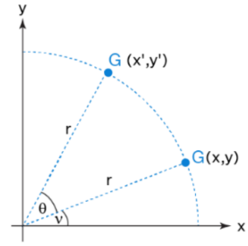
$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

Gambar 2.5. Matriks Rotasi pada Bidang 2D

Arah rotasi yang dilakukan di matriks ini berlawanan arah jarum jam.

Derivasi formula matriks rotasi

Untuk mendapatkan bentuk matriks pada gambar 2.5., dilakukan penurunan formula dengan menggambar sebuah vektor di bidang kartesius, misal vektor G. Vektor G mempunyai elemen (x, y), kemudian vektor G tersebut dirotasi berlawanan arah jarum jam dengan sudut sebanyak θ. Elemen vektor G yang baru adalah (x', y')[3].



Gambar 2.6. Penurunan Matriks Rotasi 2D (sumber dari [3])

Transformasi vektor G bisa ditulis dalam bentuk matriks rotasi sebagai berikut:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = R(\theta) \begin{bmatrix} x \\ y \end{bmatrix}$$

Gambar 2.7. Persamaan Umum Matriks Rotasi

Untuk mendapatkan nilai R(θ), representasikan elemen (x, y) dan (x', y') ke koordinat polar menjadi:

$$x = rcos(v) \qquad (1)$$

$$y = rsin(v) \qquad (2)$$

$$x' = rcos(v + \theta) \qquad (3)$$

$$y' = rsin(v + \theta) \qquad (4)$$

Dengan identitas trigonometri, persamaan (3) dan (4) dapat disederhanakan menjadi:

$$x' = rcos(v)\cos(\theta) - r\sin(v)\sin(\theta) \qquad (5)$$

$$y' = r\sin(v)\cos(\theta) + rcos(v)\sin(\theta) \qquad (6)$$

Kemudian persamaan (5) dan (6) disubstitusikan dengan (1) dan (2) menjadi:

$$x' = x\cos(\theta) - y\sin(\theta) \qquad (7)$$

$$y' = y\cos(\theta) + x\sin(\theta) \qquad (8)$$

Bentuk persamaan (7) dan (8) bisa diubah ke matriks, didapatkan persamaan:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Gambar 2.8. Persamaan Matriks Rotasi di Bidang 2D

C. Analisis Pergerakan Menggunakan Matriks Rotasi

Penentuan sudut rotasi

Pada gim Holocure, sudut rotasi digunakan dalam menargetkan musuh atau bergerak ke item yang drop di tanah. Misal dengan salah satu senjata yaitu *Holo Bomb*:

Gambar 2.9. Senjata Peledak : *Holo Bomb*

Gim akan menghitung sudut antara karakter dan posisi musuh terdekat yang ada, (x1, y1) dan (x2, y2). Perhitungan sudut bisa dihitung dengan menggunakan formula:

$$\theta = \text{arctan2}(y2 - y1, x2 - x1)$$

Gambar 2.10. Rumus Penentuan Sudut Rotasi

Perhitungan ini membuat penargetan dengan presisi yang tepat, bahkan ketika musuh bergerak secara dinamis.

Penggunaan fungsi arctan2 adalah perluasan dari fungsi tan^{-1} untuk menghindari masalah pembagian nol dan memberikan hasil yang akurat pada semua kuadran.[4]

Perhitungan perubahan posisi

Perhitungan perubahan posisi juga menggunakan matriks rotasi. Biasanya diterapkan pada proyektil atau trap. Contohnya salah satu proyektil pada gim Holocure, yaitu *Glowstick* dan *Psycho Axe*:

(a) (b)

Gambar 2.11. Senjata Proyektil: (a) *Glowstick, (b) PsychoAxe*

Saat *Glowstick* dilempar, lintasannya ditentukan menggunakan matriks rotasi pada Gambar 2.5. Ini membuat proyektil dapat bergerak di lintasan yang telah diperhitungkan, membuatnya dapat mengenai target walaupun tidak berada di garis yang lurus.

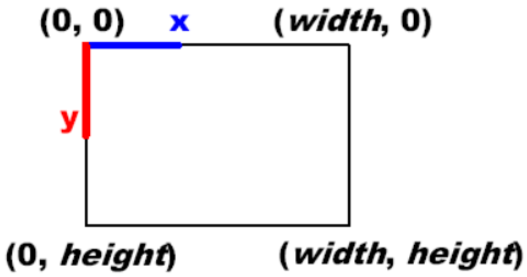
D. Representasi Karakter dalam Koordinat 2D

Posisi karakter dalam koordinat

Dalam sistem koordinat 2D, posisi setiap karakter direpresentasikan sebagai titik (x, y) pada bidang kartesius.

Sistem koordinat pada gim 2D berbeda dengan sistem koordinat matematika konvensional. Titik awal (0, 0) pada sistem koordinat gim 2D berada pada pojok kiri atas, dengan nilai x positif ke kanan dan nilai y positif itu ke bawah.

Perbedaan dalam sistem koordinat ini dapat diperhitungkan selama kita menyadarinya. Ingatlah bahwa gambar 2D dalam gim selalu memiliki sumbu y yang turun dari atas, bukan naik dari bawah.



Gambar 2.12. Sistem Koordinat dalam Gim 2D (sumber dari [5])

Titik pusat rotasi karakter

Setiap karakter atau objek memiliki titik pusat rotasi atau biasa disebut *pivot point* yang menjadi acuan saat transformasi rotasi dilakukan.

Titik pusat ini umumnya diposisikan di tengah *sprite* (gambar dua dimensi) atau karakter untuk menciptakan efek rotasi yang natural.

Saat objek berotasi, titik pusat ini akan berada di posisi yang sama, sedangkan bagian lainnya akan mengelilingi titik pusat tersebut.

(a) (b) (c)

Gambar 2.13. Titik Pusat Rotasi pada Karakter

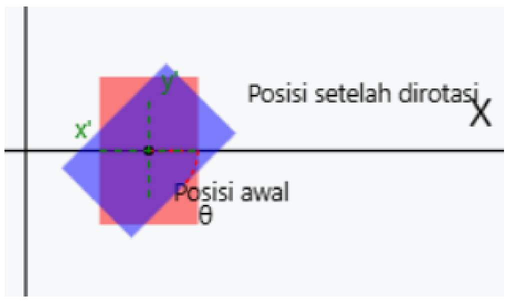
Dari gambar 2.13. bisa dilihat bahwa garis merah menunjukkan arah rotasi dan titik pusat rotasi berada di ujung garis merah di dalam segitiga.

Transformasi koordinat dalam rotasi

Pada proses transformasi rotasi, terdapat perubahan koordinat yang melibatkan dua sistem referensi: koordinat global dan koordinat lokal.

Koordinat global mengacu pada sistem koordinat layar secara keseluruhan, yaitu posisi setiap objek diukur relatif terhadap titik acuan tetap, biasanya sudut kiri atas layar pada Gambar 2.12.

Koordinat lokal merepresentasikan posisi relatif suatu titik terhadap titik pusat objek. Misal dalam konteks objek persegi panjang, titik pusat persegi panjang akan dianggap sebagai titik awal(0, 0) dalam sistem koordinat lokal.



Gambar 2.14. Transformasi Koordinat Saat Rotasi

III. Implementasi Dan Percobaan

A. Lingkungan Implementasi

Beberapa *tools* yang digunakan untuk mengimplementasikan matriks rotasi pada penelitian ini di antaranya adalah python sebagai bahasa pemrograman dan pygame sebagai *library* visualisasi grafis. Setelah itu, IDE yang dipakai adalah VS Code, OS Windows 11, resolusi tampilan 800x600 piksel dengan frame rate 60 *fps*. Pemilihan ini didasarkan atas kemudahan penggunaan dan sesuai kebutuhan.

```python
# Constants
SCREEN_WIDTH = 800
SCREEN_HEIGHT = 600
FPS = 60
ORBIT_RADIUS = 150
ORBIT_SPEED = 2
```

Gambar 3.1. Lingkungan Implementasi Percobaan

B. Implementasi Matriks Rotasi

Implementasi matriks rotasi dibagi menjadi tiga mode utama:

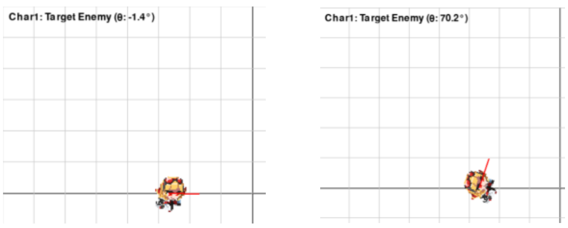
Rotasi mengikuti target

Mode ini mensimulasikan karakter yang menghadap ke arah tertentu dengan mouse cursor. Implementasi ini menggunakan fungsi arctan2 untuk menghitung sudut antara karakter dengan target.

```python
def rotate_to_point(self, target_x, target_y):
    """Rotate character to face target using arctan2"""
    dx = target_x - self.x
    dy = target_y - self.y

    self.angle = math.degrees(math.atan2(-dy, dx))
    self.update_sprite()
```

Gambar 3.2. Kode Implementasi Rotasi Mengikuti Target



(a) (b)

Gambar 3.3. Percobaan Rotasi Mengikuti Target: (a) Target Berada di Kanan Karakter Membentuk Sudut -1.4 Derajat (b) Target Berada di Atas Kanan Karakter Membentuk Sudut 70.2 Derajat

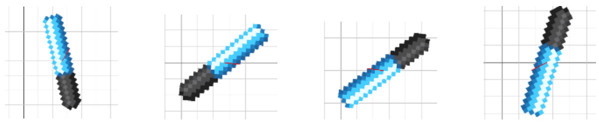
Perhitungan sudut ini menggunakan selisih koordinat x dan y antara posisi karakter dan target, yang kemudian dikonversi ke dalam derajat.

Rotasi di tempat

Mode ini mensimulasikan karakter yang berotasi di tempat dengan kecepatan konstan. Implementasi ini menggunakan *increment* sudut yang tetap setiap *frame*, dengan nilai sudut berada pada rentang ($0^o - 360^o$). Mode ini berguna untuk mensimulasikan objek yang berputar seperti senjata dalam gim Holocure.

```python
def update_sprite(self):
    """Update sprite rotation"""
    self.sprite = pygame.transform.rotate(self.original_sprite, self.angle)
    self.rect = self.sprite.get_rect(center=(self.x, self.y))
```

Gambar 3.4. Kode Implementasi Rotasi di Tempat



(a) (b) (c) (d)

Gambar 3.5. Percobaan Rotasi di Tempat Dengan Berbagai Sudut Rotasi: (a) (b) (c) (d)

Rotasi orbit

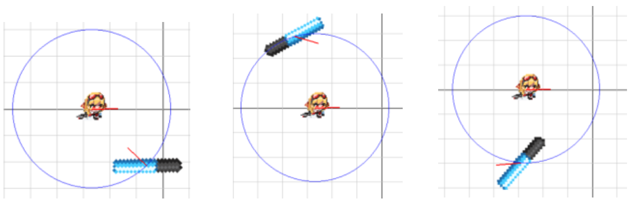
Mode ini mensimulasikan objek yang bergerak dalam pola melingkar dengan karakter sebagai titik pusatnya.

```python
def orbit_around(self, center_char):
    """Orbital movement pattern for weapons"""
    self.orbit_angle = (self.orbit_angle + ORBIT_SPEED) % 360
    self.x = center_char.x + ORBIT_RADIUS * math.cos(math.radians(self.orbit_angle))
    self.y = center_char.y + ORBIT_RADIUS * math.sin(math.radians(self.orbit_angle))
    self.angle = self.orbit_angle + 90
    self.update_sprite()
```

Gambar 3.6. Kode Implementasi Rotasi Orbit

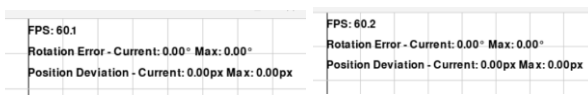
Mode ini menggunakan fungsi trigonometri sin dan $cosin$ untuk menghitung posisi karakter pada setiap frame, dengan radius orbit yang tetap. Implementasi ini juga mencakup penyesuaian orientasi karakter agar selalu menghadap arah geraknya.

.



(a) (b) (c)

Gambar 3.7. Percobaan Implementasi Rotasi Orbit Dengan Karakter Sebagai Titik Pusat: (a) (b) (c)



Gambar 3.8. Performa Sistem Selama Pengujian

C. Skenario Pengujian

Sistem rotasi diuji melalui serangkaian skenario yang dirancang untuk memverifikasi keakuratan implementasi matriks rotasi. Pengujian mode rotasi mengikuti target difokuskan pada keakuratan penghitungan sudut dengan menggunakan rumus arctan2 dan daya tanggap sistem terhadap perubahan posisi target. Dilakukan penghitungan waktu respons sistem dan deviasi sudut dari arah yang diinginkan.

Konsistensi kecepatan rotasi dan stabilitas posisi karakter diuji untuk mode rotasi di tempat. Seperti contoh pada Gambar 3.5, pengujian dilakukan pada tiga titik untuk memverifikasi keakuratan sudut rotasi. Kestabilan karakter saat berotasi juga diamati untuk memastikan bahwa tidak ada gerakan yang tidak diinginkan.

Tiga aspek utama - konsistensi radius orbit, konsistensi kecepatan sudut, dan konsistensi orientasi karakter - difokuskan saat menguji mode orbit. Gambar 3.7 memvisualisasikan lintasan yang dihasilkan, menunjukkan bagaimana karakter mempertahankan jarak yang konstan dari pusat orbit sambil mempertahankan orientasinya.

D. Hasil Pengujian dan Pembahasan

Implementasi matriks rotasi telah berhasil mencapai tingkat akurasi dan performa yang diinginkan. Pada mode rotasi mengikuti target, program menunjukkan responsif yang baik dalam mengikuti pergerakan *pointer mouse*, dengan rotasi karakter yang lancar dan presisi.

Pengujian mode rotasi di tempat mengkonfirmasi stabilitas sistem yang baik. Karakter dapat melakukan rotasi dengan kecepatan konstan tanpa adanya pergeseran posisi yang signifikan dari titik awalnya. Pergerakan rotasi berjalan dengan *smooth* dan konsisten. Ini menunjukkan keberhasilan implementasi matriks rotasi dalam mempertahankan posisi sambil melakukan transformasi rotasi.

Mode rotasi orbit juga menunjukkan performa yang cukup baik, dengan karakter berhasil mempertahankan jarak orbit yang konsisten dari titik pusat. Pergerakan orbital berjalan lancar dengan kecepatan *angular* yang terjaga konstan, dan orientasi karakter tetap akurat mengikuti arah pergerakannya. Hal ini membuktikan bahwa penggunaan fungsi trigonometri dalam implementasi berhasil menciptakan pergerakan orbital yang stabil.

Keseluruhan hasil pengujian memvalidasi efektivitas implementasi matriks rotasi dalam menciptakan berbagai pola pergerakan yang dibutuhkan dalam gim 2D. Kombinasi dari ketiga mode rotasi memberikan fleksibilitas yang tinggi dalam menghasilkan pergerakan karakter yang dinamis dan responsif, sambil tetap mempertahankan presisi dan performa yang optimal.

IV. ANALISIS

Dari hasil pengujian dan implementasi yang telah dilakukan, penerapan matriks rotasi dalam gim 2D memperlihatkan efektivitas yang tinggi dalam membuat pergerakan karakter yang presisi. Pada implementasi rotasi mengikuti target seperti yang ditemui pada mekanisme Holo Bomb, penerapan dengan formula arctan2 sangat efektif dalam menentukan sudut antara karakter/objek dan target. Hasil pengukuran menunjukkan bahwa sistem bisa mencapai keakuratan sangat tinggi, dengan error kurang dari 1 derajat, yang memungkinkan penargetan yang tepat dalam konteks gameplay Holocure.

Implementasi gerakan rotasi orbital, seperti yang diterapkan pada proyektil Glowstick pada Gambar 3.7., berhasil menciptakan pola serangan area yang konsisten dan dapat diprediksi.

Kestabilan sistem dihitung dengan deviasi minimal radius orbital dan kecepatan sudut yang konstan, ini memberikan kemampuan untuk memahami dan memanfaatkan pola pergerakan senjata secara efektif. Ini juga bisa dikembangkan lagi dengan menambahkan beberapa variasi pola orbital seperti gerakan bumerang dengan orbital elips yang memberikan area serangan yang lebih luas.

V. KESIMPULAN

Implementasi dan analisis yang telah dilakukan dalam penelitian ini menunjukkan bahwa penerapan matriks rotasi berperan penting dalam menciptakan sistem pergerakan karakter yang akurat dan efisien dalam gim 2D seperti Holocure. Penggunaan formula matematika yang tepat, meliputi arctan2 untuk penargetan dan fungsi trigonometri untuk pergerakan orbital, terbukti mampu menghasilkan pergerakan yang presisi dengan error minimal.

Berdasarkan hasil pengujian, implementasi matriks rotasi berhasil menciptakan sistem yang tidak hanya akurat dalam perhitungan sudut dan posisi, tetapi juga efisien dalam penggunaan sumber daya komputasi.

Keberhasilan implementasi ini menegaskan pentingnya transformasi matriks sebagai komponen fundamental dalam pengembangan gim modern, memberikan dasar matematis yang kuat untuk menciptakan pengalaman bermain yang menarik dan menantang bagi pemain.

Implementasi dan percobaan yang dilakukan pada makalah ini pun masih bisa dikembangkan lagi dengan menambahkan lebih banyak variasi objek ataupun dengan variasi pola rotasi yang lebih luas dan unik.

VI. UCAPAN TERIMA KASIH

Penulis ingin mengucapkan rasa syukur kepada Tuhan Yang Maha Esa karena berkat rahmat dan kasih-Nya, penulis dapat menyelesaikan tugas makalah aljabar linier dan geometri dengan tepat waktu.

Penulis juga ingin mengucapkan rasa terima kasih kepada kedua orang tuanya yang selalu mendukung jiwa raganya dan memberikan motivasi dalam menjalankan perkuliahan. Penulis tidak lupa untuk mengucapkan rasa terima kasih juga kepada dosen-dosen pengampuh mata kuliah aljabar linier dan geometri, Bapak Ir. Rila Mandala, M. Eng atas bimbingan belajar dan pengajaran selama 1 semester kelas aljabar linier dan geometri.

Terakhir, penulis ingin mengucapkan rasa terima kasih kepada teman-temannya yang menjadi sahabat perjuangan sebagai mahasiswa Teknik Informatika ITB 2023.

REFERENSI

[1] 3Blue1Brown. “Linear Transformations and Matrices | Essence of Linear Algebra, Chapter 3.” YouTube, 7 Aug. 2016, www.youtube.com/watch?v=kYB8IZa5AuE. Accessed 30 Dec. 2024.

[2] 3Blue1Brown. “Matrix Multiplication as Composition | Chapter 4, Essence of Linear Algebra.” *www.youtube.com*, 9 Aug. 2016, www.youtube.com/watch?v=XkY2DOUCWMU. Accessed 30 Dec. 2024.
[3] “Rotation Matrix - Definition, Formula, Derivation, Examples.” *Cuemath*, 2021, www.cuemath.com/algebra/rotation-matrix. Accessed 1 Jan. 2025.
[4] “Arctan vs Arctan2.” *Geosciences LibreTexts*, 5 Aug. 2020, geo.libretexts.org/Courses/University_of_California_Davis/GEL_056%3A_Introduction_to_Geophysics/Geophysics_is_everywhere_in_geology.../zz%3A_Back_Matter/Arctan_vs_Arctan2. Accessed 1 Jan. 2025.
[5] Whitaker, R. B. “Monogame - Introduction to 2D Graphics - RB Whitaker’s Wiki.” *Wikidot.com*, 2025, rbwhitaker.wikidot.com/monogame-introduction-to-2d-graphics. Accessed 1 Jan. 2025.
[6] Jorge Rodriguez. “Math for Game Developers - Character Movement 8 (Matrices).” *YouTube*, 19 Apr. 2013, www.youtube.com/watch?v=8sqv11x10lc. Accessed 1 Jan. 2025.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 27 Desember 2024

Nama dan NIM
Muhammad Hazim Ramadhan Prajoda / 13523009