

Aplikasi Transformasi Lanjar dalam Permainan *Dragon Nest*

Michael - 13514108
Program Studi Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13514108@std.stei.itb.ac.id

Abstract— Aljabar geometri banyak digunakan dalam video game, terutama dalam memanipulasi grafik pada *video game*. Dalam makalah ini, akan dibahas aplikasi dari salah satu materi dalam aljabar geometri, yaitu transformasi lanjar dalam permainan *Dragon Nest*. Transformasi lanjar dalam *Dragon Nest* digunakan pada pergerakan kursor, gerakan karakter, serangan, serta *zoom in* dan *zoom out*.

Keywords— *transformasi lanjar, dragon nest, kursor, zoom*

I. PENDAHULUAN

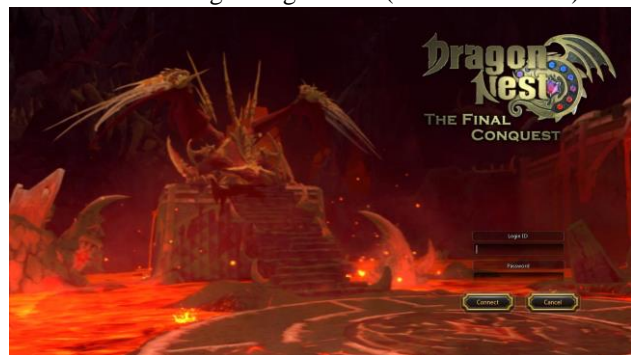
Di zaman modern ini, kehidupan manusia tidak dapat dipisahkan dari teknologi. Selain teknologi, kehidupan dari sebagian besar orang juga tidak dapat dipisahkan dari internet. Segala aspek kehidupan sudah dapat dikaitkan dengan teknologi dan internet, contohnya berjualan saat ini sudah dapat dilakukan secara *online* pada situs berjualan, mengobrol dengan teman dapat menjadi mudah dengan menggunakan jejaring sosial, mengetahui keadaan dunia saat ini juga dapat dilakukan dengan mudah dengan berselancar di web. Semuanya dapat dilakukan dengan mudah menggunakan komputer dan telepon genggam yang dilengkapi dengan internet. Salah satu aspek yang tidak terpisahkan dari teknologi dan internet adalah permainan. Saat ini, sudah banyak *video game* yang dapat dimainkan secara *online* bersama orang lain walaupun orang tersebut berada pada kota maupun negara yang berbeda. Salah satu *game online* yang banyak dimainkan orang adalah *Dragon Nest*.

Pembuatan *video game*, baik *video game offline* maupun *online* tidak terlepas dari kebutuhkannya berbagai ilmu dalam bidang informatika, salah satunya aljabar geometri. Salah satu materi dalam aljabar geometri yang banyak diimplementasikan pada *video game* adalah transformasi lanjar. Selain transformasi lanjar, sebenarnya materi aljabar geometri seperti vektor, kuaternion, dan perkalian geometri juga banyak digunakan dalam pembuatan *video game*, terutama untuk bagian grafis.

II. DRAGON NEST



Gambar 1. Logo Dragon Nest (dnsea.wikia.com)



Gambar 2. Layar login pada *Dragon Nest*

Dragon Nest adalah sebuah *video game* MMORPG (*Massive Multiplayer Online Role Playing Game*). *Dragon Nest* dikembangkan oleh perusahaan Eyedentity dan dipasarkan oleh berbagai perusahaan di berbagai negara. Perusahaan yang memasarkan *game* *Dragon Nest* di Indonesia adalah Gemscool. Selain server Indonesia, banyak pemain Indonesia yang bermain *Dragon Nest* dengan server yang menampung pemain-pemain Asia Tenggara. *Dragon Nest* server Asia Tenggara tersebut dipasarkan oleh perusahaan Cherry Credits.

Dragon Nest dimainkan dengan mengendalikan seorang karakter yang dibuat oleh pemain. *Game* ini dimainkan secara *third person*, artinya pemain dapat melihat karakter yang sedang dimainkannya. Di tengah layar ditampilkan sebuah *crosshair* yang menunjukkan arah gerak karakter dan tempat serangan karakter akan dilancarkan.

Karakter yang dimainkan dapat berinteraksi dengan karakter pemain lain, lalu dapat pergi ke *dungeon* atau *nest*. *Dungeon* adalah tempat seperti hutan dan

semacamnya yang terdapat *monster*. Karakter yang menyelesaikan *dungeon* akan mendapat *experience* yang akan menjadi parameter untuk naik level. *Nest* adalah *dungeon* yang lebih sulit dan memberi *experience* yang lebih banyak dan barang yang lebih bagus dari *dungeon* biasa.

Selain fitur utamanya yaitu menjelajahi *dungeon* dan *nest*, *Dragon Nest* juga memiliki fitur-fitur pelengkap lain yaitu pemain dapat memasak, memancing dan berkebun. Selain itu, pemain juga dapat masuk ke sebuah tempat festival yang memiliki berbagai macam permainan. Tujuan dari fitur-fitur pelengkap itu adalah mendapatkan barang-barang yang akan berguna untuk menjelajahi *dungeon* atau *nest*.

III. DASAR TEORI

A. Vektor

Vektor adalah sebuah kuantitas atau fenomena yang terdiri atas dua komponen, yaitu besar dan arah. Vektor dituliskan dengan menggunakan huruf kecil bercetak tebal atau dengan huruf kecil dengan lambang vektor di atasnya.

Contoh : \mathbf{v}, \vec{v}

Komponen-komponen pada vektor di R^n dapat dituliskan dengan dua cara, yaitu dengan dipisahkan menggunakan koma atau dengan menggunakan matriks.

Contoh : $\mathbf{v} = (v_1, v_2, \dots, v_n), \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$

B. Transformasi Linier

Transformasi linier adalah transformasi dari suatu ruang vektor (domain) ke ruang vektor lain (kodomain). Misalkan V dan W adalah ruang vektor, maka $T: V \rightarrow W$ adalah transformasi linier dari ruang vektor V ke ruang vektor W , jika untuk semua vektor \mathbf{u} dan \mathbf{v} di dalam ruang vektor V , kedua hal ini berlaku :

1. $T(\mathbf{u}+\mathbf{v}) = T(\mathbf{u}) + T(\mathbf{v})$
2. $T(k\mathbf{u}) = kT(\mathbf{u})$ (k adalah skalar)

Beberapa transformasi khusus :

1. Misalkan V dan W adalah ruang vektor, jika $T: V \rightarrow W$ didefinisikan sebagai $T(\mathbf{v}) = 0$ untuk semua \mathbf{v} anggota ruang vektor V Maka T adalah transformasi linier yang disebut transformasi nol (*zero transformation*)
2. Misalkan V adalah ruang vektor, jika $T: V \rightarrow V$ didefinisikan sebagai $T(\mathbf{v}) = \mathbf{v}$ untuk semua \mathbf{v} anggota ruang vektor V Maka T adalah transformasi linier yang disebut transformasi identitas (*identity transformation*) dari V .

Beberapa properti dari transformasi linier :

1. $T(0) = 0$

2. $T(-\mathbf{v}) = -T(\mathbf{v})$ untuk semua \mathbf{v} anggota ruang vektor V
3. $T(\mathbf{u}-\mathbf{v}) = T(\mathbf{u})-T(\mathbf{v})$ untuk semua \mathbf{u}, \mathbf{v} anggota ruang vektor V
4. Jika $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$ adalah vektor-vektor di dalam ruang vektor V dan k_1, k_2, \dots, k_n adalah skalar, maka: $T(k_1\mathbf{v}_1+k_2\mathbf{v}_2+\dots+k_n\mathbf{v}_n) = k_1T(\mathbf{v}_1) + k_2T(\mathbf{v}_2) + \dots+k_nT(\mathbf{v}_n)$

Kernel (ruang null) dari transformasi linier T , dituliskan sebagai $\ker(T)$ adalah himpunan vektor \mathbf{v} dalam ruang vektor V yang dipetakan T ke 0 .

$$\ker(T) = \{\mathbf{v} \in V : T(\mathbf{v}) = 0\}$$

Jangkauan (*range* atau jelajah) adalah semua vektor di dalam kodomain W yang merupakan bayangan oleh transformasi linier T dari paling sedikit satu vektor di dalam domain V .

$$\text{range}(T) = \{\mathbf{w} \in W : \mathbf{w} = T(\mathbf{v}) \text{ untuk beberapa } \mathbf{v} \in V\}$$

T dikatakan isomorfis jika T adalah transformasi satu ke satu dan transformasi pada.

T isomorfis jika $\ker(T) = 0$ dan $\text{range}(T) = W$.

Sebuah transformasi linier dari R^n ke R^m dapat dinyatakan dengan sebuah matriks berukuran $m \times n$ yang disebut sebagai matriks transformasi.

Transformasi linier dari R^n ke R^m :

Misalkan i_1, i_2, \dots, i_n adalah basis standar untuk R^n dan A adalah matriks $m \times n$ yang berisi $T(i_1), T(i_2), \dots, T(i_n)$ sebagai kolom dari matriks A . Secara umum:

$$T(i_1) = \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} \quad T(i_2) = \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix} \quad T(i_n) = \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

A adalah matriks transformasi standar dari T .

Kombinasi transformasi :

Jika T_1, T_2, \dots, T_3 adalah transformasi:

$$T_1(\mathbf{x}) = A_1\mathbf{x}$$

$$T_2(\mathbf{x}) = A_2\mathbf{x}$$

$$T_k(\mathbf{x}) = A_k\mathbf{x}$$

Jika T_1, T_2, \dots, T_k dilakukan berturut-turut, maka matriks transformasi gabungannya adalah

$$A = A_k A_{k-1} \dots A_2 A_1$$

IV. APLIKASI TRANSFORMASI LINIER DALAM DRAGON NEST

1. Pergerakan Kursor dan Karakter

Posisi kursor dapat berubah secara rotasi saat *mouse* digerakkan ke kiri dan ke kanan. Pada saat *mouse* digerakkan seperti itu, karakter akan mengubah arah pandangannya. Dengan demikian, kursor yang menjadi penanda arah depan dari

karakter juga berubah. Perubahan tersebut dilakukan dengan cara transformasi linier. Transformasi linier dilakukan terhadap vektor yang menjadi representasi dari lokasi kursor. Vektor tersebut adalah vektor R^3 , yang menandakan lokasi kursor secara kartesian.

Pergerakan mouse sejauh sekian millimeter akan menyebabkan karakter berotasi sejauh sebesar derajat. Rasio rotasi per pergerakan mouse dapat diatur oleh pemain dengan mengubah *mouse sensitivity*. Semakin besar *mouse sensitivity*, semakin besar derajat rotasi karakter untuk gerakan mouse yang sama.



Gambar 3. Mouse Sensitivity

Matriks transformasi linier yang menyebabkan kursor berotasi adalah:

$$R_i(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix}$$

$$R_j(\alpha) = \begin{bmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{bmatrix}$$

$$R_k(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$R_i(\alpha)$ adalah rotasi sepanjang sumbu x (x tetap) sebesar α , $R_j(\alpha)$ adalah rotasi sepanjang sumbu y (y tetap) sebesar α , dan $R_k(\alpha)$ adalah rotasi sepanjang sumbu z (z tetap) sebesar α .

Namun, jika kita menginginkan karakter kita berotasi sejauh n derajat, karakter tidak akan tiba-tiba berputar sebesar n derajat, namun karakter akan berotasi perlahan-lahan. Karakter akan berotasi sebesar, misalnya 1 derajat (angka eksak tidak diketahui karena harus melihat langsung kode program) sebanyak sekian kali hingga total perputarannya sebesar n derajat.

Gerakan mouse kiri kanan menyebabkan karakter berotasi dengan sumbu vertikal tetap, berarti rotasinya adalah rotasi sepanjang sumbu y, dengan menggunakan matriks transformasi R_j . Misalkan kita ingin merotasi karakter sebesar 30 derajat, maka matriks transformasi totalnya adalah:

$$R_j(30) = \begin{bmatrix} \cos 1 & 0 & \sin 1 \\ 0 & 1 & 0 \\ -\sin 1 & 0 & \cos 1 \end{bmatrix} \begin{bmatrix} \cos 1 & 0 & \sin 1 \\ 0 & 1 & 0 \\ -\sin 1 & 0 & \cos 1 \end{bmatrix} \dots$$

sebanyak 30 matriks



(a)



(b)

Gambar 4. (a) Karakter dan kursor sebelum rotasi (b) Karakter dan kursor setelah rotasi ke arah kanan

Selain gerakan mouse kiri kanan yang menyebabkan karakter berotasi ke kiri dan ke kanan, gerakan mouse ke atas dan ke bawah juga menyebabkan kursor berotasi. Karakter tetap diam di tempat, namun pandangannya bergerak ke atas atau ke bawah. Konsep rotasi tetap sama dengan rotasi kiri kanan, namun ada batasan derajat rotasi, karena karakter diam dan mata karakter hanya dapat melihat hal yang ada di depan karakter.

Rotasi atas bawah adalah rotasi dengan sumbu horizontal tetap, berarti rotasi sepanjang sumbu x dengan matriks transformasi R_i .



(a)



(b)

Gambar 5. (a) Batas rotasi ke atas (b) Batas rotasi ke bawah

Kita juga dapat melakukan rotasi dalam semua sumbu sekaligus jika menggerakkan mouse dalam arah diagonal. Jika menggerakkan mouse dalam arah diagonal, maka matriks transformasi yang digunakan adalah hasil perkalian matriks dari R_j dan R_i .

$$R_d(\alpha, \beta) = R_i(\alpha)R_j(\beta)$$

2. Memperbesar dan Memperkecil Serangan

Beberapa serangan dalam *Dragon Nest* awalnya memiliki ukuran yang kecil, namun lama-kelamaan, ukuran serangan itu akan semakin membesar. Pembesaran ukuran itu dilakukan dengan transformasi lanjar.

Matriks transformasi lanjutannya adalah:

$$A = \begin{bmatrix} S_i & 0 & 0 \\ 0 & S_j & 0 \\ 0 & 0 & S_k \end{bmatrix}$$

S_i adalah skala perbesaran untuk vektor arah sumbu x, S_j adalah skala perbesaran untuk vektor arah sumbu y, dan S_k adalah skala perbesaran untuk vektor arah sumbu z. Untuk perbesaran benda 3 dimensi, jika diinginkan hasil perbesarannya simetris dengan benda awal, maka $S_i = S_j = S_k$.

Vektor yang ditransformasi adalah selisih vektor-vektor ujung dari benda yang diperbesar dengan vektor yang menjadi pusat benda (vektor yang ditransformasi adalah vektor ukuran benda dihitung dari pusatnya). Kemudian, setelah ditransformasi, vektor-vektor ujung tersebut diganti dengan vektor hasil transformasi.

Sama seperti rotasi, perbesaran benda juga tidak dilakukan secara tiba-tiba. Jika ingin memperbesar benda 2 kali lipat, tidak serta-merta menggunakan matriks transformasi lanjar dengan $S_i = S_j = S_k = 2$, namun dilakukan dengan cara memperbesar sebesar misalnya $S_i = S_j = S_k = 1.1$ sebanyak x kali hingga perbesaran mencapai 2 kali lipat.

$$S(2) = \begin{bmatrix} 1.1 & 0 & 0 \\ 0 & 1.1 & 0 \\ 0 & 0 & 1.1 \end{bmatrix} \begin{bmatrix} 1.1 & 0 & 0 \\ 0 & 1.1 & 0 \\ 0 & 0 & 1.1 \end{bmatrix} \dots \text{sebanyak}$$

7 kali.

Salah satu contoh serangan yang membesar di permainan *Dragon Nest* adalah skill *Big Mecha Bomb* milik class *Gear Master*. Awalnya, karakter akan mengeluarkan sebuah bom kecil dan dalam waktu

beberapa detik, jika bom tidak dilempar, bom tersebut akan membesar.



(a)



(b)

Gambar 6. (a) Ukuran awal *Big Mecha Bomb* (b) Ukuran akhir *Big Mecha Bomb*

3. Zoom In dan Zoom Out

Zoom in dan *zoom out* dilakukan dengan cara yang sama dengan memperbesar dan memperkecil serangan, yaitu dengan transformasi lanjar yang matriks transformasinya sama dengan matriks transformasi 2.

Pada saat melakukan *zoom in*, ukuran objek benda yang ditampilkan membesar, dengan cara mentransformasi vektor-vektor benda sehingga benda menjadi membesar. *Zoom in* dan *zoom out* pada *Dragon Nest* juga memiliki batas maksimum.



(a)



(b)

Gambar 7. (a) Batas zoom in di *Dragon Nest* (b) Batas zoom out di *Dragon Nest*

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, 12 Desember 2015

Michael - 13514108

V. KESIMPULAN

- Rotasi karakter dan kursor, baik dalam arah kiri-kanan maupun atas-bawah dalam permainan *Dragon Nest*, dilakukan dengan menggunakan transformasi lanjar.
- Pengaturan ukuran objek, baik saat *zoom in* dan *zoom out* maupun saat melakukan skill atau serangan, dilakukan dengan transformasi lanjar.
- Transformasi lanjar yang dilakukan pada *Dragon Nest* untuk satu perintah tidak hanya satu kali, namun berulang-ulang sehingga hasil animasi yang didapatkan menjadi mulus.
- Selain dalam *Dragon Nest*, kemungkinan besar transformasi lanjar yang digunakan untuk pergerakan secara rotasi dan untuk pengaturan ukuran objek digunakan juga oleh game-game lainnya, terutama game bergenis RPG dan MMORPG seperti *Dragon Nest*.
- Selain transformasi lanjar, desain grafis pada permainan *Dragon Nest* juga berhubungan erat dengan materi aljabar geometri lainnya, seperti vektor dan operasi geometri.

REFERENCES

- [1] <http://www.math.ucsd.edu/~jlobue/102/graphics.pdf>, diakses pada tanggal 5 Desember 2015, pukul 10.02
- [2] <https://www.math.ku.edu/~mandal/math290/m290NotesChSIX.pdf>, diakses pada tanggal 5 Desember 2015, pukul 13.04
- [3] <http://linear.ups.edu/html/section-LT.html>, diakses pada tanggal 5 Desember 2015, pukul 13.08
- [4] <http://blogs.msdn.com/b/rezanour/archive/2011/10/02/math-primer-series-matrices-ii-linear-transformations.aspx>, diakses pada tanggal 5 Desember 2015, pukul 13.12
- [5] <http://whatis.techtarget.com/definition/vector>, diakses pada tanggal 5 Desember 2015, pukul 14.02
- [6] <http://dn.cherrycredits.com/>, diakses pada tanggal 8 Desember 2015, pukul 9.41
- [7] <http://dnsea.wikia.com/>, diakses pada tanggal 8 Desember 2015, pukul 9.45