

## DAFTAR SINGKATAN

GPU = *Graphical Processing Unit*

VR = *Virtual Reality*

3D = *Dimensi Tiga*

PRNG= *Pseudorandom Number Generator*

PCG = *Procedural Content Generator*

PRNG PCG = *Pseudorandom Number Generator Permuted Congruential Generator*

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Industri permainan merupakan salah satu industri yang menjanjikan dimasa sekarang [1]. Permainan simulasi yang baik dapat digunakan untuk memperkecil kesalahan yang terjadi di dunia nyata. Pekerjaan seperti pilot, astronot, dan personil militer dapat memperkecil kecelakaan dan cedera [2]. Permainan-permainan masa sekarang banyak menggunakan *Procedural Content Generator* untuk membuat permainannya. Algoritma PCG digunakan untuk membuat elemen-elemen pada permainan, terutama permainan-permainan yang bertemakan petualangan atau *endless exploration* seperti *No Man Sky* [3]. Dengan menggunakan PCG, kita bisa membuat *level* berdasarkan *skill* atau pengalaman pemain [4], membuat *level* desain agar lebih dinamis [5], atau membuat permainan yang *level*-nya tidak pernah sama. Dengan semakin beragamnya sebuah *level*, dalam simulasi maka dapat dipastikan bahwa setiap kesalahan yang terjadi dan mungkin terjadi dapat diantisipasi dengan menggunakan kemungkinan konten-konten yang tidak terduga setiap kali permainan dijalankan [2]. Dengan menggunakan keberagaman konten latihan, atlet-atlet panahan dapat mengembangkan kemampuannya lebih luas lagi, karena dengan menggunakan *Procedural Content Generator* kita bisa membuat *level* dengan konten yang sangat acak [5]. Masalah yang ditemui adalah menggunakan *Procedural Content Generator* untuk membuat dan mengacak konten yang ada. *Procedural Content Generator* biasanya menggunakan *Pseudorandom Number Generator* untuk membuat angka “acak”. Salah satu PRNG yang umum digunakan adalah *Linear Congruential Generator* atau LCG. Pada LCG, diketahui bahwa suatu angka memiliki periode tertentu, dimana angka akan berulang pada iterasi tertentu. Hal tersebut menyulitkan apabila pengembang menghindari terjadinya pola berulang pada *Procedural Content Generator* yang ingin kita buat. Beberapa PRNG yang memiliki data nilai tinggi secara statistik memiliki masalah performa yang kurang cepat dalam perhitungan angka acak.

Salah satu solusi adalah menggunakan *Random Generator* yang baik secara statistik dan juga secara performa. *Permuted Congruential Generator* merupakan solusi kedua masalah tersebut. *Permuted Congruential Generator* merupakan PRNG yang dikembangkan untuk mengatasi masalah mod  $2^n$  pada LCG [6].

## 1.2 Rumusan Masalah

Rumusan Masalah dari penyusunan Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana *level* dan target dimunculkan secara acak?
2. Bagaimana cara implementasi *Permuted Congruential Generator* untuk memposisikan elemen yang ada dalam setiap *level*?
3. Apakah konten yang dibuat dapat di manipulasi untuk menyesuaikan suasana lingkungan yang ada?
4. Seberapa besar kemungkinan dua atau lebih objek yang saling bersinggungan?
5. Bagaimana membuat *level* dan konten dengan *Prosedural Content Generator*?

## 1.3 Tujuan

Tujuan dari penyusunan Tugas Akhir ini adalah sebagai berikut:

1. Membuat sebuah permainan panahan yang dapat di implementasikan ke *Virtual Reality*.
2. Dapat membuat *level* dan target dalam permainan panahan muncul secara acak.
3. Manipulasi konten dengan menyesuaikan suasana lingkungan yang ada.
4. Menghitung persentasi objek yang bersinggungan dan mengurangi masalah *error* pada pengacakan target dan *Agent Spawn*.
5. Membedakan penggunaan algoritma *Linear Congruential Generator (LCG)* dengan algoritma *Combined Linear Congruential Generator (CLCG)* dan *Permuted Congruential Generator (PCG)*.

#### **1.4 Batasan Masalah**

- 1) Algoritma yang digunakan pada tugas akhir ini adalah *Permuted Congruential Generator*.
- 2) Sistem yang akan dibuat dalam bentuk aplikasi permainan 3D.
- 3) Metode yang digunakan untuk menciptakan *Level* merupakan *Procedural Content Generator* dengan algoritma *Permuted Congruential Generator*.
- 4) *Level* yang ada merupakan Design penuh menggunakan *Procedural content generator*.

Input yang di berikan berupa *seed*, *increment* dan *multiplier* untuk melakukan perhitungan di PCG

#### **1.5 Sistematika Penulisan TA**

Pembuatan Buku Tugas Akhir ini di susun secara struktural, berikut adalah sistematikanya:

##### **BAB I PENDAHULUAN**

Pada BAB I Memiliki sistematik berupa latar belakang penelitian ini dilakukan, rumusan masalah, batasan masalah, metodologi penelitian dan sistematika penulisan.

##### **BAB II KAJIAN PUSTAKA**

Pada BAB II ini berisi tentang penjelasan apa itu *Congruential Generator Algorithm*, *Unity 3D*, teori metode lain sebagai pembanding dan *Virtual Reality*.

##### **BAB III ANALISIS DAN PERANCANGAN**

Pada BAB III menjelaskan gambaran umum sistem yang dibangun, teori metode yang digunakan dan skenario pengujian.

##### **BAB IV IMPLEMENTASI DAN PENGUJIAN**

Pada BAB IV ini akan melakukan pengujian pada sistem yang sudah dibuat berupa kepada Unity 3D.

## **BAB V KESIMPULAN DAN SARAN**

Pada BAB V ini berisi kesimpulan dari penelitian Tugas Akhir ini dan saran untuk menunjang penelitian yang akan dilakukan selanjutnya

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Game**

Permainan atau *game* adalah kata berbahasa Inggris yang berarti permainan atau pertandingan, atau bisa diartikan sebagai aktifitas terstruktur yang biasanya dilakukan untuk bersenang-senang, dengan aturan tertentu sehingga ada yang menang dan ada yang kalah, biasanya dalam konteks tidak serius atau dengan tujuan refreshing. Akan tetapi tidak semua *game* hanya untuk bersenang-senang melainkan ada juga yang untuk melatih kemampuan seperti olahraga panahan dan olahraga lainnya .

Dalam permainan panahan dituntut untuk konsentrasi tinggi dan kesabaran selain itu menurut hasil kajian yang dilakukan oleh *US National Electronics Injury Surveillance System (NEISS)* menunjukkan bahwa kegiatan olahraga panahan dipandang sebagai salah satu cabang olahraga yang memiliki tingkat keamanan yang tinggi dibandingkan beberapa jenis cabang olahraga lainnya. Olahraga panahan juga sebagai cara belajar yang digunakan dalam menganalisa interaksi antara sejumlah pemain maupun perorangan yang menunjukkan strategi-strategi yang rasional.

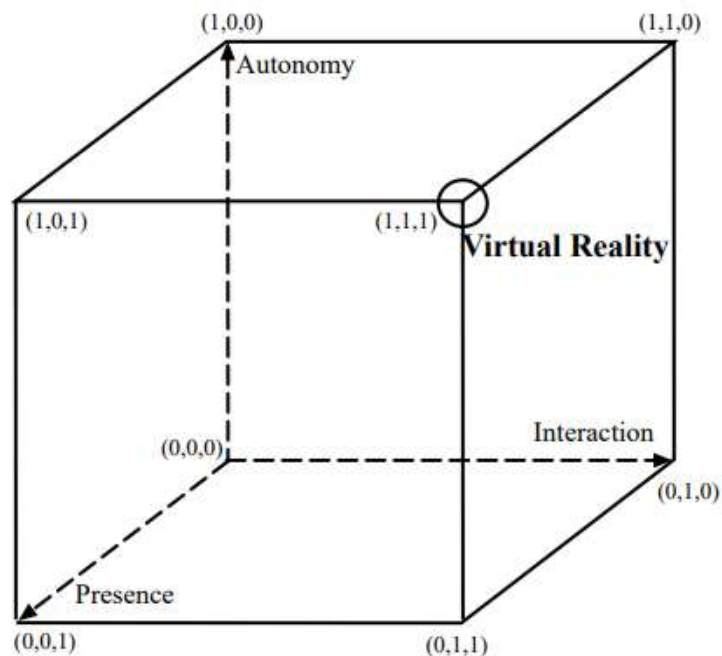
#### **2.2 Virtual Reality**

Perkembangan teknologi *Virtual Reality (VR)* yang cepat memberikan berbagai dampak baik mulai dari dukungan model manusia dan lingkungan realitas. *Virtual Reality* dapat meningkatkan karakteristik melalui model manusia dengan teknologi realitas, dimana teknologi tersebut dapat menghindari kecelakaan dan kehilangan yang mungkin tidak perlu terjadi. Data yang diambil dan dikumpulkan dari sistem penangkapan gerak berupa data yang asli tetapi memiliki kekurangan yaitu interaksi dan pengendalian. Pengambilan data tersebut juga dapat digunakan sebagai analisis

penelitian yang nantinya akan mempengaruhi gerak yang digunakan didalam *Virtual Reality*.

Ada beberapa pendapat mengenai *Virtual Reality* diantaranya:

- Menurut H. Fuch “*Real-time interactive graphics with three-dimensional models, combined with a display technology that gives the user the immersion in the model world and direct manipulation*” [23].
- Menurut M. Gigante “*The illusion of participation in a synthetic environment rather than external observation of such an environment. VR relies on a three-dimensional, stereoscopic head-tracker displays, hand/body tracking and binaural sound. VR is an immersive, multi-sensory experience*” [24].
- Menurut Jargon “*Computer simulations that use 3D graphics and devices such as the DataGlove to allow the user to interact with the simulation.*” [25].



Gambar 2.1. Penggambaran VR di dimensi tiga menurut D.zelter [26].

Gambar merupakan penggambaran *Virtual Reality* menurut D.Zelter. menurut D. Zelter semua VR bersifat interaktif dan memiliki pengalaman *imersif* di dunia simulasi.

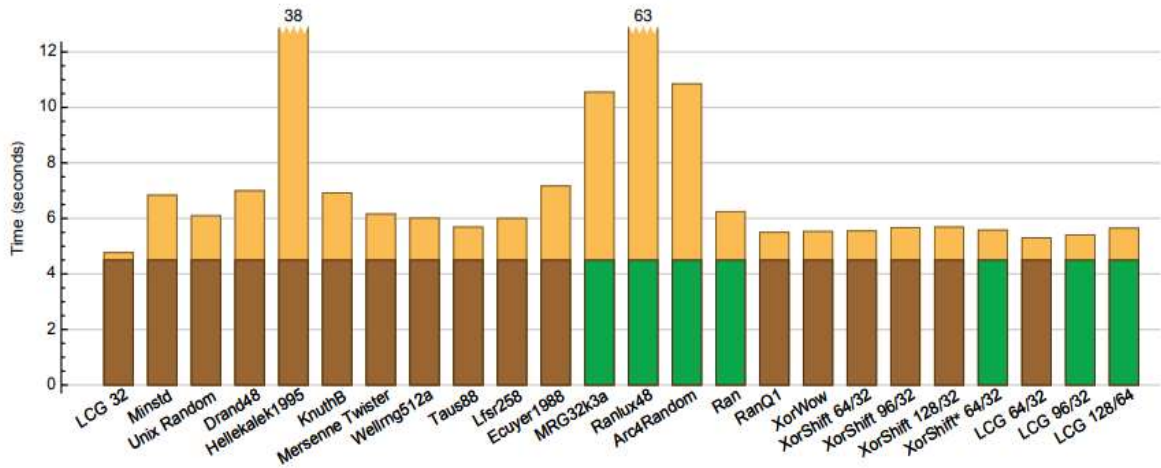
### **2.3 Objek Dimensi tiga**

Objek dimensi tiga merupakan objek yang memiliki tiga buah kordinat yaitu x, y dan z untuk mendefinisikan sebuah objek, koordinat tersebut juga berpengaruh untuk setiap rotasi yang ada di dalam objek tersebut. Karena manusia hidup di ruang dimensi tiga, maka menggunakan objek dimensi tiga simulasi akan mampu dibuat menyerupai dunia nyata. Pada penelitian ini setiap konten dan ruangan yang ada akan di acak menggunakan *Pseudorandom Number Generator* dengan metode *Permuted Congruential Generator* pada setiap vektor yang mendefinisikan posisi dari objek tersebut. Setiap *range* pengacakan akan dibatasi dengan panjang lebar dan tinggi dari suatu objek tersebut, sehingga pengacakan yang terjadi masih masuk akal.

### **2.4 Pseudorandom Number Generator**

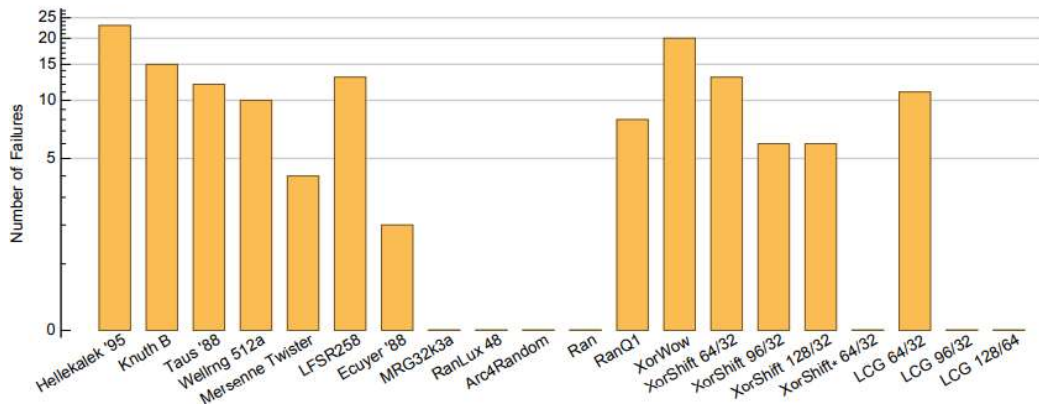
*Pseudorandom Number Generator* (PRNG) merupakan algoritma yang menghasilkan deret bit yang unik yang berasal dari nilai awal yang di sebut dengan *seed* [12]. *Pseudorandom number generator* digunakan dalam banyak hal seperti, permainan, robotik, peningkatan performa, modeling dan simulasi, algoritma genetik dan masih banyak lagi. Dengan menggunakan PRNG angka dapat di *generate* secara acak berdasarkan *seed* yang diberikan. Sebuah PRNG harus melalui beberapa tes untuk menguji keacakan dan performa dari *random number* tersebut. TestU01 merupakan tes yang digunakan untuk menguji *Empirical Statistik Testing* pada PRNG jenis *Uniform* [6]. *Empirical Statistik Testing* merupakan suatu tes yang digunakan untuk menentukan apakah *Random Number Generator* memproduksi angka acak yang berperilaku sebagaimana angka acak berperilaku. Pada gambar dan gambar ditampilkan data performa dan keacakan PRNG yang sering digunakan.





Gambar 2.2. Performa beberapa PRNG menghadapi test Small Crush [6].

Pada gambar diatas grafik yang bertanda hijau menandakan bahwa PRNG tersebut telah berhasil mengatasi test BigCrush dan Crush.



Gambar 2.3. Hasil test crush dan BigCrush PRNG yang sering digunakan [6].

## 2.5 Unity3d

*Unity3d* adalah sebuah *game engine* yang berbasis *cross-platform*. *Unity3d* dapat digunakan untuk membuat sebuah *game* yang bisa digunakan pada perangkat komputer, ponsel pintar android, iPhone, PS3, dan bahkan X-BOX. *Unity3d* adalah sebuah *tool* yang terintegrasi untuk membuat *game*, arsitektur bangunan dan simulasi. *Unity3d* bisa untuk *games PC* dan *games online*. Untuk *games online*

diperlukan sebuah *plugin*, yaitu *Unity Web Player*, sama halnya dengan *Flash Player* pada *Browser*. *Unity3d* tidak dirancang untuk proses desain atau modelling, dikarenakan *unity* bukan *tool* untuk mendesain. Banyak hal yang bisa dilakukan dengan *unity3d*, ada fitur *audio reverb zone*, *particle effect*, dan *Sky Box* untuk menambahkan langit. Fitur *scripting* yang disediakan, mendukung tiga bahasa pemrograman, *JavaScript*, *C#*, dan *Boo*.

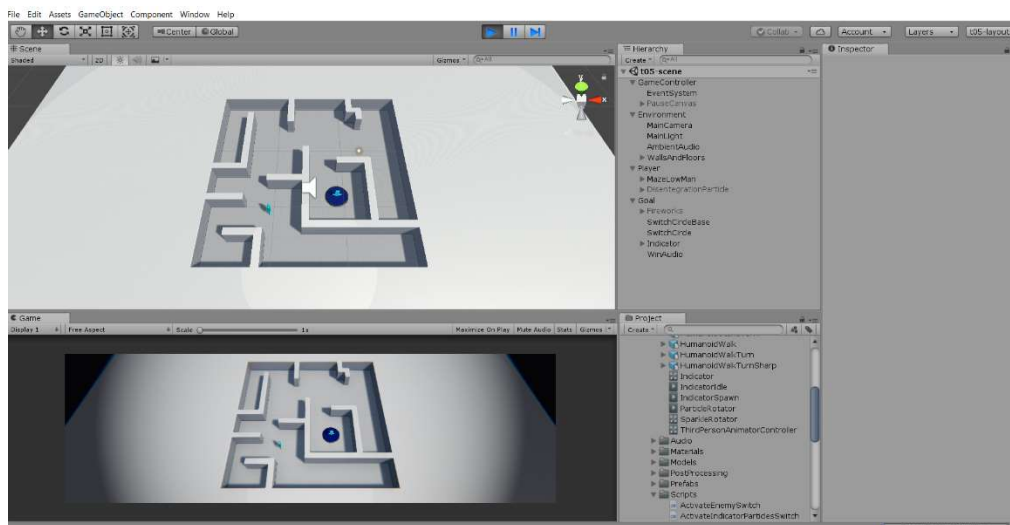
## 2.6 Sejarah *Unity3d*

*Unity3d* adalah mesin lintas platform yang dikembangkan oleh *Unity3d Technologies*. Lahir pada tahun 2005 sebagai alat pengembangan untuk *game* di *OSX* (Sistem Operasi *Apple*), sekarang *Unity3d* merupakan salah satu alat yang paling banyak digunakan dalam pengembangan *video game*. Dikembangkan dalam *C / C ++*, mendukung kode yang ditulis dalam *C-sharp*, *Javascript* dan *Boo* yang memungkinkan untuk digunakan secara bersamaan dalam proyek yang sama. *Unity3d* memungkinkan pengembangan *video game* yang dapat berfungsi di *PlayStation 3*, *Xbox 360*, *IOS* (sistem operasi seluler *Apple*), *Android*, *PC Windows*, *Linux*, *OSX*, dan aplikasi web. *Unity3d* memiliki perangkat internal sendiri untuk pembangunan lingkungan permainan video virtual, tetapi juga mendukung model yang diimpor dari program lain seperti *3DStudioMax*, *Maya*, *Blender*, dan banyak lainnya. *Unity3d* juga mendukung *physics engine* yang dikenal luas *NVIDIA PhysX* [5].

Banyaknya komunitas dari seluruh dunia yang menawarkan bantuan dan saran serta berkontribusi untuk terus meningkatkan perangkat lunak ini. Filosofi *Unity3d* didasarkan pada prinsip-prinsip berikut:

- *Scene*: lokasi virtual tempat *game* tersebut diatur. Adegan bisa menjadi *level* atau satu bagian dari *scene* tersebut.
- *Assets*: Blok paling dasar yang membentuk *video game*. Ini adalah sumber daya yang digunakan untuk membuat *game* seperti tekstur, skrip, dan XXX.
- *Game object*: Objek aktif ditampilkan dalam suatu adegan, yaitu untuk membuat yang kosong (yang hanya memiliki kemungkinan untuk bertransformasi menjadi dirotasi atau dipindahkan), atau untuk memilih dari yang sudah dibuat sebelumnya.

- *Prefabs*: Terkadang pengembang perlu menggunakan kembali objek permainan yang sama di banyak adegan dan berkali-kali. *Prefabs* memungkinkan memorisasi objek lengkap agar mereka dapat digunakan kembali di lain waktu.
- *Components*: objek tertentu yang dapat dikaitkan dengan objek permainan untuk mengubah perilaku mereka. Fungsionalitas dapat ditambahkan dan penampilan mereka dapat diubah. *Components* umum dalam *Unity3d* adalah pemancar partikel, kamera, dan lampu.



Gambar 2.4. Tampilan *Unity3d* [6]

- *Scene*: Bagian yang memungkinkan untuk menambah, memindahkan, atau menghapus objek aktif dalam suatu *scene*. Dengan kata lain, ini adalah editor visual di *Unity3d*.
- *Game*: Di sini seluruh permainan video dapat dipratinjau melalui kamera utama (jika ada). Saat mode uji aktif, seluruh permainan dapat diuji, meskipun efek yang lebih baik tidak aktif.
- *Hierarchy*: Ini mencantumkan semua objek aktif yang ada dalam pemandangan sebenarnya. Melalui daftar ini, Anda dapat dengan cepat memilih objek atau menambahkan aktivitas atau komponen tertentu ke objek aktif.

- *Project: Vision global* dari file proyek. Dari sini, dimungkinkan untuk memilih aktivitas dan komponen yang diimpor dari program lain, atau diunduh dari sumber lain.
- *Inspector*: Setelah memilih sesuatu dalam tampilan *Hierarchy* atau proyek, di bagian ini memungkinkan untuk melihat dan memodifikasi propertinya. Jika skrip dipilih, memungkinkan untuk melihat kode tertulis di bagian dalam. Dari sini juga dimungkinkan untuk menambahkan sesuatu ke objek yang dipilih.

## 2.7 C#

*Script* harus dilampirkan ke *GameObject* untuk dipanggil oleh *Unity3d*. *Script* ditulis dalam bahasa khusus yang dapat dimengerti. Dan, melalui bahasa inilah kita dapat berbicara dengan mesin dan memberikan instruksi kita. Bahasa yang digunakan dalam *Unity3d* disebut C # (diucapkan *C-sharp*). Semua bahasa yang dioperasikan *Unity3d* adalah bahasa *Script* yang berorientasi objek. Seperti Bahasa apapun, bahasa *scripting* memiliki sintaks, atau bagian dari ucapan, dan bagian-bagian utama disebut variabel, fungsi, dan kelas.

## 2.8 Permuted Congruential Generator

*Permuted Congruential Generator* merupakan salah satu teknik algoritma dalam menghasilkan PRNG. *Permuted Congruential Generator*, ditemukan pada tahun 2014. *Permuted Congruential Generator* menggunakan Algoritma LCG (*Linear Congruential Generator*) yang dikembangkan sehingga meningkatkan sifat Modulo-2<sup>n</sup> pada LCG [6]. *Permuted Congruential Generator* memiliki data statistik yang baik, kode yang relatif cepat dan sederhana, dan juga ukuran *state* yang kecil. PCG berhasil melewati TestU01 *BigCrush* dengan kemungkinan 2<sup>38</sup> untuk tipe PCG-XSH-RR (*Permuted Congruential Generator - XorShifted - Random Rotation*), dimana bit yang dibutuhkan untuk lulus tes tersebut sebanyak 2<sup>35</sup>. *Permuted Congruential Generator* menggunakan *Unsigned integer* untuk memperluas distribusi dari *range number* yang dimiliki. *Permuted Congruential Generator* tidak mengalami pengulangan angka dalam siklus tertentu, sehingga secara statistik *Permuted Congruential Generator* dapat digunakan untuk mencapai tingkat keacakan yang tinggi.

Berikut dapat dituliskan dengan sederhana:

$$output = rotate32((state \wedge (state \gg 18)) \gg 27, state \gg 59) \quad (1)$$

## 2.9 Procedural Content Generator (PCG)

*Procedural Content Generator* adalah metode untuk membuat konten *game* yang pertama kali diperkenalkan oleh kelompok yang dikenal sebagai *demoscene* sekitar 20 tahun yang lalu. *Procedural Content Generator* merupakan suatu cara untuk membangun suatu konten dalam permainan dengan menggunakan Algoritma tertentu [9]. *Procedural Content Generator* bisa menggunakan AI (*Artificial Intelligence*) atau manusia untuk membuat konten didalam permainan [11]. Dengan menggunakan *Procedural Content Generator*, pengembang permainan dapat menggantikan beberapa designer dan artis dengan algoritma-algoritma tertentu, dan permainan dapat diproduksi lebih cepat dan murah serta tetap menjaga kualitas dari setiap *level* yang dibuat [10]. Dengan menggunakan *Procedural Content Generator*, pengembang permainan dapat membuat konten dalam permainan secara cepat dan tetap mengikuti aturan-aturan dan pola dalam permainan design. *Procedural Content Generator* juga dapat membuat *level design* dalam permainan menjadi lebih dinamis sehingga permainan yang menggunakan *Procedural Content Generator* dapat dibuat mengikuti dari perkembangan dari pemainnya. Dengan menggunakan *Procedural Content Generator*, pengembang dapat memanipulasi ruang dimensi tiga atau dimensi dua, yang artinya setiap objek dalam permainan dapat direkayasa rotasi objek, ukuran objek dan lokasi objek yang ada. Pada gambar merupakan contoh hasil PCG, *enemy* di *spawn* sesuai dengan banyaknya koin yang dikumpulkan.



Gambar 2.5. PCG di Supermario [9]

Dengan menggunakan *Permuted Congruential Generator* sebagai PRNG dan *procedural Content Generator* yang membuat konten berdasarkan urutan dari indeks konten maka dapat dipastikan bahwa konten yang di berikan bisa berubah-ubah dengan tingkat keacakan yang tinggi. Gambar 2.6 merupakan proses pembangunan konten yang ada dalam permainan tersebut.



Gambar 2.6. PCG yang merekam koordinat saat tombol loncat ditekan [9].

## 2.10 Standar Deviasi

Standard deviasi atau  $\sigma$  merupakan ukuran untuk mengukur banyaknya variasi atau penyebaran dari suatu data nilai. Nilai standar deviasi apabila kecil maka mengindikasikan penyebaran data yang ada dekat menuju rata-rata. Rumus standar deviasi adalah.

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=0}^n (x_i - \bar{x})^2} \quad (2)$$

Dimana  $x_i$  merupakan nilai sample ke  $i$ ,  $\bar{x}$  merupakan rata-rata dan  $n$  merupakan banyak data.

### 2.11 Mean

*Mean* adalah nilai dari menjumlahkan semua data dan membaginya dengan banyak data yang tersedia berikut ini merupakan rumus *mean*:

$$\bar{x} = \frac{1}{n} \sum_{i=0}^n x_i \quad (3)$$

Dimana  $x_i$  merupakan nilai sample ke  $i$ ,  $\bar{x}$  merupakan rata-rata dan  $n$  merupakan banyak data.

### 2.12 Median

*Median* adalah nilai tengah dari suatu sampel distribusi berdasarkan nilai tertinggi dan terendah dari sampel tersebut. Rumus median dibagi dua yaitu:

- Data ganjil

$$Q_2 = X_{\frac{n}{2}+1} \quad (4)$$

- Data genap

$$Q_2 = \frac{X_{\frac{n}{2}} + X_{\frac{n}{2}+1}}{2} \quad (5)$$

Dimana  $X$  merupakan data ke- dan  $n$  merupakan panjang data.

### 2.13 Standard Error

*Standard error* atau *Standard error of the mean* adalah nilai yang mencerminkan keakuratan dari sample data yang diambil. Berikut ini merupakan rumus standard error:

$$SE = \frac{\sigma}{\sqrt{n}} \quad (6)$$

Dimana  $\sigma$  merupakan rata-rata dan  $n$  merupakan banyak data.

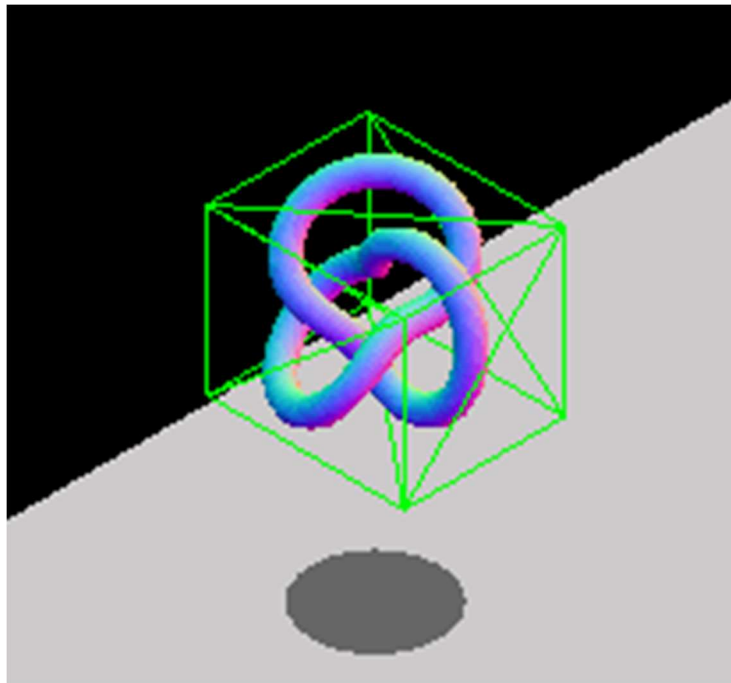
### 2.14 Sample Variance

*Sample variance* merupakan sampel ukuran seberapa jauh sekumpulan bilangan tersebar. Berikut ini merupakan rumus *sample variance*:

$$S^2 = \frac{\sum(X-\bar{X})}{n-1} \quad (6)$$

Dimana  $\bar{X}$  merupakan rata-rata,  $n$  merupakan banyak data dan  $X$  merupakan nilai sampel.

### 2.15 AABBs (*axis-aligned bounding boxes*) Collision detection



Gambar 2.7. Limas segitiga dengan bounding box.