

PROPOSAL TUGAS AKHIR

GAME 2D PLATFORMING “Ctrl+It” MENGGUNAKAN FINITE STATE MACHINE BERBASIS UNITY C#

Nama : David Brave Navy Putra
NRP : 218116715
Jurusan/Prodi/Major : Teknik Informatika / S1 / *Computational Intelligence*
Dosen Pembimbing : Hendrawan Armanto, S.Kom, M.Kom
Co-Pembimbing : -

I. Latar Belakang

Perkembangan teknologi informatika atau disiplin ilmu komputer berperan sangat penting bagi kehidupan manusia. Dengan adanya perkembangan teknologi terutama di bidang informatika, manusia dapat dengan mudah mendapatkan, memberi juga menyimpan informasi dan melakukan kegiatan sehari-hari dengan bantuan teknologi yang ada. Teknologi membawa manusia melihat lebih jauh ke dunia luar, membuka wawasan berpikir, serta membangun kreativitas untuk menciptakan hal-hal dan inovasi baru. Karena teknologi dapat melakukan banyak hal yang berada di luar kemampuan manusia seperti menyelesaikan perhitungan matematis berjuta-juta dalam satuan waktu detik dan juga tidak akan terjadi *human error*. Teknologi yang berkembang sendiri dapat berbentuk benda nyata (*hardware*) maupun berbentuk ide, atau sistem (*software*).

Salah satu bentuk perkembangan teknologi informatika yang sedang booming pada masa ini adalah game. Game adalah media hiburan yang dapat digunakan oleh setiap kalangan yang untuk menghilangkan rasa jenuh atau bosan. Game juga memiliki manfaat lain seperti pengembangan otak, melatih pemecahan masalah, meningkatkan konsentrasi, melatih kecepatan koordinasi mata dengan tangan, dan lainnya. Terdapat banyak sekali genre pada game, beberapa di antaranya adalah *action*, *shooter*, *horror*, *strategy*, *role playing*, *sport*, *vehicle simulations*, *platforming*, *construction management*, *adventure*, *artificial life* dan *puzzle games*.

Pada awal tahun 2000-an, industri game terutama permainan digital mengalami perkembangan yang sangat pesat. Hal ini ditunjukkan dengan banyaknya perangkat keras untuk bermain game yang terus bermunculan dan terus berkembang, seperti *mobile phone*, tablet, PC (komputer), laptop, *console* (Playstation, XBOX, dan Nintendo Wii), dan lain-lain. Oleh karena itu, game ini akan dibuat dengan Unity Game Development Tools atau yang lebih dikenal dengan Unity yang merupakan salah satu *game engine* paling populer dalam pembuatan game dengan target utama pengguna PC atau laptop.

II. Tujuan

Pada subbab ini akan menjelaskan mengenai tujuan-tujuan yang akan dihasilkan dalam pembuatan tugas akhir ini. Tujuan dari tugas akhir ini adalah sebagai berikut:

1. Membuat game 2D menggunakan Unity game engine terutama Unity Real-Time Development Platform dan Unity Game Development Tools.
2. Merancang game 2D dengan konsep *platformer* untuk desktop atau laptop berbasis unity, yang menarik dan dapat menghibur para pemainnya.
3. Mengaplikasikan AI menggunakan *Finite State Machine* agar NPC terasa lebih hidup.

III. Teori Penunjang

Terdapat juga beberapa teori yang menunjang pembuatan aplikasi game ini, seperti:

A. Object Oriented Programming (OOP)

Pemrograman berorientasi objek merupakan metode untuk membuat program menggunakan *class* dan *object*. OOP membuat pengembangan dan pemeliharaan menjadi lebih mudah. OOP menyediakan beberapa konsep seperti *inheritance*, *data binding*, *polymorphism*, *encapsulation*, dan lain-lain.

B. Unity

Pembuatan aplikasi game akan menggunakan Unity Game Development Tools sebagai media utama pembuatan *character*, *terrain*, *level*, animasi dan *code* yang akan dituliskan pada Unity GameObject Script, menggunakan Bahasa C#.

C. C#

C# merupakan bahasa *native* Unity, dan juga Bahasa yang dikembangkan Microsoft sendiri. Dengan meningkatnya penggunaan Windows, C# sendiri menjadi semakin terkenal, sehingga saat ini, C# dapat digunakan pada semua *development platform*, mulai dari desktop, web, console, hingga mobile. Pada Unity, C# digunakan untuk memberikan *script* pada sebuah GameObject agar *object* tersebut dapat berinteraksi dengan *object* atau user *input*.

D. Artificial Intelligence (AI)

Kecerdasan buatan atau *artificial intelligence* (AI) merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan atau berpikir seperti dan sebaik manusia. Pada game, AI bertindak dalam mengendalikan karakter komputer agar dapat bermain dengan pemain manusia maupun melawan pemain manusia. Penggunaan AI pada game *platformer* adalah sebagai penentu pergerakan apa yang akan dilakukan NPC atau musuh.

E. AI State Machine

Agar pergerakan AI terlihat lebih natural dan masuk akal, dibutuhkan metode untuk merancang perilaku musuh atau NPC. Metode yang akan digunakan adalah *Finite State Machine* (FSM). Selain untuk sistem kontrol, FSM adalah model yang umum digunakan untuk merancang perilaku AI pada sebuah game yang memiliki kelebihan berupa kecepatan komputasinya dan kemudahan dalam pemahaman implementasi. Penentuan diagram *state* sesuai dengan jenis perilaku yang sudah ditentukan, dengan

kondisi-kondisi tertentu yang dapat memperlancar alur permainan dan pergerakan AI.

F. Photo Editor

Mengingat game yang dibuat adalah 2D game, maka akan diperlukan aplikasi yang dapat membuat, dan mengedit file *sprite* 2D yang akan digunakan di dalam game. Sehingga dibutuhkan sebuah aplikasi *photo editor*, untuk menghasilkan *sprite* dan *background terrain* yang dapat dipakai. Photo editor yang akan dipakai adalah Adobe Photoshop CC 2018. Photoshop adalah perangkat lunak pengedit citra buatan Adobe Systems yang dikhususkan untuk pengeditan foto/gambar dan pembuatan efek. Photoshop melakukan penyuntingan berbasis pixel, sehingga lebih cocok digunakan dalam pembuatan *sprite*.

**Teori penunjang dapat berubah atau bertambah dalam tahap pengembangan program untuk memenuhi kebutuhan dan perkembangan teknologi terkini yang ada.*

IV. Ruang Lingkup

Ruang lingkup yang akan dibahas pada proposal tugas akhir ini adalah hal-hal sebagai berikut:

4.1 Story Game

Storyline dari game ini akan mengambil konsep tentang adanya kehidupan pada dimensi lain. Apa jadinya jika hidup seorang pejuang yang ingin menyelamatkan desanya dari sebuah monster ganas. Monster tersebut sudah meneror seluruh orang-orang yang dia sayangi selama berbulan-bulan. Namun pejuang tersebut tidak memiliki kemampuan dan kekuatan yang cukup untuk melawan monster tersebut. Sang pejuang tersebut akhirnya berusaha mencari bantuan untuk mengalahkan monster tersebut.

Namun bagaimana jika bantuan tersebut datang bukan dari orang di sekitarnya? Bukan guru bela diri, atau kesatria perkasa. Namun orang yang berada 1 dimensi di atas pejuang tersebut. Pemain game inilah orang yang bisa membantu pejuang tersebut. Pemain yang ada pada 3 dimensi, diminta membantu pejuang yang hidup di 2 dimensi dengan menggunakan cara berpikir dan *object* yang ada pada dimensinya sendiri.

Object tersebut bisa berupa *key* pada keyboard player yang ditaruh untuk membantu pejuang tersebut berjalan, maupun *input* mouse yang membantu pejuang tersebut mengalahkan musuh yang ada. Atau pun membantu pejuang tersebut mengakali musuh yang ada. Hingga akhirnya pejuang dengan bantuan pemain dapat mengalahkan monster yang sudah meneror desa itu selama berbulan-bulan.

4.2 Arsitektur Game

Berikut adalah bagian-bagian desain dan arsitektur yang akan dibuat dalam Tugas Akhir pembuatan game ini.

a. Deskripsi Game

Game ini adalah sebuah game *adventure platformer* yang terinspirasi dari game klasik Super Mario Bros. Di mana pada game ini, pemain bisa secara langsung membantu karakter utama untuk menyelesaikan misi dan level yang tersedia. Bantuan tidak sekadar cara berjalan namun juga cara mengalihkan atau mengalahkan musuh. Cakupan bantuan yang diberi pemain jauh lebih besar dan banyak dibandingkan game Super Mario Bros, mengingat tingkat kesulitan *puzzle*, level maupun AI dibuat lebih berkembang dan kompleks. Game ini diharapkan bisa memberi rasa senang yang pernah dikenang pemain game pendahulunya, namun dengan *gameplay* dan tampilan yang lebih menarik menggunakan metode juga *tools* yang lebih baik dan modern.

b. Visual

Elemen visual dari game ini berupa 2D *picture* dan 2D *sprite*. *Sprite* tersebut akan dibuat oleh penulis dengan model *pixel art*. Mengingat keterbatasan kemampuan seni penulis, beberapa *asset* akan dilakukan *outsourcing*, kepada pihak lain yang memiliki kemampuan lebih dalam hal ini atau akan diambil dari web *resource* seperti Unity Asset Store dengan pemberian *credit* sesuai ketentuan jika *asset* digunakan.

c. Non-Player Character (NPC)

NPC pada game ini akan memiliki AI state machine untuk membuat NPC dapat terasa lebih hidup. Setiap state machine bisa memiliki jumlah dan tipe AI *state* yang berbeda beda. NPC yang ada dapat dibagi menjadi dua, yaitu *neutral*, dan *aggressive*.

Untuk NPC yang *neutral*, maka *state machine* berisi *wandering*, *idle* dan *flee*, dengan *player action* berupa *interact*. Sedangkan untuk NPC *aggressive*, *state machine* dapat berupa *patrol*, *idle*, *search*, *wander*, *seek*, *arrive* dan *attack*. *Action player* dapat dibuat menyerang untuk NPC yang bertipe *aggressive*.

Walau tipe NPC sama, *state machine* bisa saja berbeda. Seperti NPC musuh yang menyerang dengan pedang harus berjalan menuju pemain dahulu, baru menyerang saat sampai di depan pemain. Sedangkan musuh yang memiliki serangan proyektil, bisa tidak perlu menuju pemain namun dapat langsung menembak.

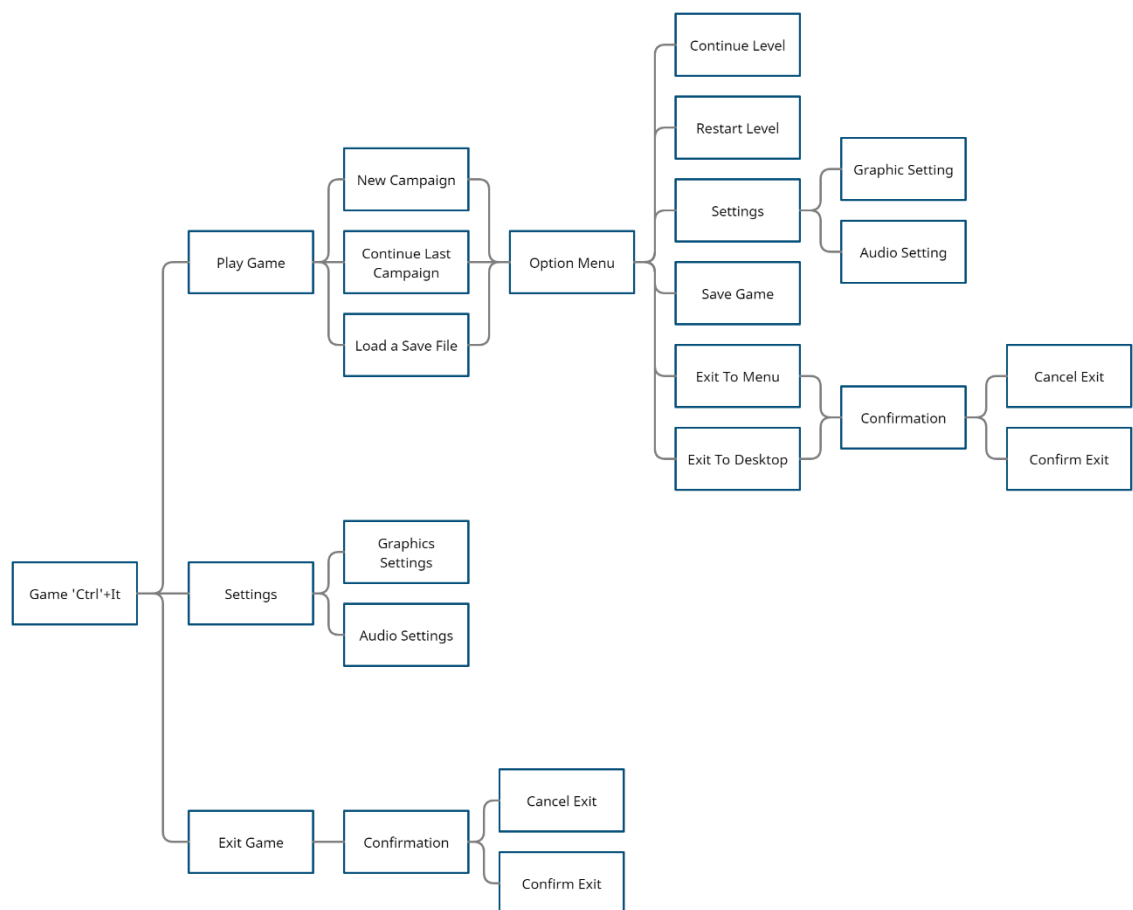
d. Skenario

Setelah cerita ditampilkan, maka muncul layar permainan. Pemain akan masuk ke level pertama untuk mengenali mekanik game. Mulai dari *key* yang digunakan, juga fungsi dari *key* yang ada. Pada saat inilah *gameplay* game akan dimulai. Sehingga pemain dapat menggunakan mekanik yang telah disampaikan pada tutorial untuk menyelesaikan scenario game.

4.3 Block Diagram

Dalam aplikasi game yang akan dibuat, user pada saat membuka aplikasi akan tersedia pilihan menu mengenai *game*, *settings* dan *exit*. Jika user memilih game, maka user akan diberi pilihan untuk membuat *campaign* baru, melanjutkan *campaign* terakhir jika ada file dan melakukan *load save* file. Pada pilihan *setting*, user dapat mengatur *interface setting*. *Interface* terdiri atas 2 hal yaitu *graphical* dan *audio*. User dapat mengatur kedua hal tersebut sesuai kebutuhan. Jika user memilih untuk *exit game*, akan dimunculkan pesan konfirmasi untuk memastikan user benar-benar ingin keluar dan bukan merupakan *misclick* atau kesalahan *input*.

Berikut adalah blok diagram dari aplikasi game yang akan dibuat :



Gambar 4.1

Block Diagram Game

(Sumber gambar : Dokumen pribadi)

4.4 Gameplay

Aplikasi game yang akan dibuat adalah game yang berbasis pada PC atau *personal computer*. Game ini adalah 2D *platformer* game dengan goal

utama mengalahkan musuh yang ada, dan mencapai tujuan akhir level. Untuk mencapai tujuan akhir level, pemain harus melalui beberapa rintangan, seperti jurang, gunung, danau ataupun kondisi *terrain* lain yang menghalangi pemain untuk berjalan lebih lanjut. Pergerakan player menggunakan layout standar seperti w, a, s, d untuk berjalan dan space bar untuk melompat. Namun yang berbeda adalah semua key yang dipakai akan memiliki interface pada layer game.



Gambar 4.2

Konsep Interface Game

(Sumber gambar : Dokumen pribadi)

Interface button tersebut dapat digunakan untuk membantu player menyelesaikan halangan yang ada. *Interface button* dapat di click dan drag ke dalam map sehingga menjadi bagian dari map, atau object. Object key inilah yang dapat berinteraksi dengan *ingame model*, *enemy* ataupun *terrain* pada level. Namun jika *key* berada pada map, maka fungsionalitas dari *key* tersebut akan hilang. Sehingga walaupun user menekan *key* pada keyboard, game tidak akan merespons *input key* tersebut.



Gambar 4.3

Konsep Usable Key

(Sumber gambar : Dokumen pribadi)

Contoh pada gambar 4.3, tombol spacebar digunakan sebagai jembatan antar dua pulau. Maka pemain dapat berjalan di atas *key* spacebar tersebut, namun pemain tidak dapat melompat karena tombol spacebar sedang dipakai sebagai *object key*. *Object key* juga dapat digunakan untuk mempengaruhi *AI behaviour*. Dimisalkan terdapat sebuah AI yang agresif kepada pemain, dan AI tersebut berusaha mengejar pemain seperti gambar 4.4 dan 4.5. Hal tersebut diakibatkan *state* AI yang berubah sesuai dengan *AI State Machine* yang dibuat.



Gambar 4.4

Aggressive AI Behaviour

(Sumber gambar : Dokumen pribadi)



Gambar 4.5

Aggressive AI Behaviour

(Sumber gambar : Dokumen pribadi)

Player dapat menggunakan tombol *interface* lain untuk memanipulasi cara kerja AI. Misalkan pada gambar 4.6, pemain menggunakan tombol shift untuk menghalangi *line of sight* AI. Sehingga AI tidak mengetahui posisi pemain, dan menjadi tidak agresif. Atau seperti gambar 4.7, pemain menggunakan tombol shift sebagai penghalang jalan agar AI tidak dapat mendekati pemain. Dengan menggunakan *object key*, *state* AI yang awalnya *seek player*, dapat diubah menjadi *wander*.



Gambar 4.6

AI Without Line Of Sight

(Sumber gambar : Dokumen pribadi)



Gambar 4.7

AI Without Path to Chase

(Sumber gambar : Dokumen pribadi)

Hal ini merupakan salah satu mekanik inti untuk menyelesaikan puzzle dan rintangan yang diberikan pada *campaign* nantinya. AI juga dapat merespons dengan aksi yang berbeda jika terdapat *key-key* lain yang digunakan pada map. Mekanik itu akan digunakan hingga level terakhir. Jika level terakhir berhasil terselesaikan, maka *campaign* dianggap tamat.

**Konsep gambar hanya digunakan untuk menjelaskan mekanik game, dan bukan merupakan hasil akhir gameplay maupun tampilan dari game yang akan dibuat.*

Game ini akan mempunyai beberapa peraturan seperti :

- Input kontrol game dilakukan dengan mouse dan keyboard.
- User dapat melanjutkan *campaign* ke level selanjutnya jika level saat ini berhasil terselesaikan.
- Pemain mati jika gagal melalui obstacle, terserang musuh atau memilih untuk melakukan restart level.
- Jika pemain mati, maka akan bangkit kembali pada checkpoint yang tersedia, secara default, checkpoint adalah posisi start di awal level.
- Pemain dapat menyimpan data *campaign* untuk dilanjutkan lain waktu.

4.5 Fitur

Game ini memiliki beberapa fitur kunci yang dibuat untuk menghasilkan game yang lebih menarik dan interaktif. Fitur pada game ini dapat dibagi menjadi dua kategori utama, yaitu fitur dari game secara umum, dan fitur dari gameplay yang ada dalam permainan.

Secara umum, aplikasi game ini akan memiliki fitur sebagai berikut :

- **Play Campaign**

Dalam *play mode* ini, player dapat menjalankan *campaign story*. *Campaign* akan terdiri dari beberapa level, level awal akan mencakup *tutorial* dan pengenalan mekanik. Level selanjutnya adalah pengenalan musuh dan tipe musuh, beserta *problem solving* untuk puzzle yang diberikan. Level terakhir akan memiliki *boss battle* yang akan memerlukan mekanik yang telah diberikan kepada pemain untuk mengalahkan boss tersebut. *Campaign* dapat di *pause* dengan *option* menu, dan dapat di *save and load*.

- **Settings**

User dapat mengubah setting sesuai kebutuhan dan keinginan. Untuk membuka setting, user dapat memilih menu setting pada startup panel game. Jika user masuk ke menu setting, akan ada beberapa submenu yang dapat dipilih user. Submenu berupa video setting, dan audio setting. Sesuai Namanya, Video setting digunakan untuk mengubah tampilan grafik, seperti fullscreen atau windowed. Audio setting digunakan untuk mengubah setting suara game seperti volume *game theme*.

Terdapat beberapa fitur dari *gameplay* sendiri, fitur-fitur tersebut adalah :

- **Checkpoint**

Fitur *Checkpoint* ini disediakan agar player dapat bangkit kembali jika player melakukan kesalahan seperti terserang musuh, gagal melompati jurang ataupun memilih *restart* level. Akan disediakan beberapa *checkpoint* di dalam sebuah level. *Checkpoint* pada awal level adalah posisi awal pemain, yang kemudian akan diperbaharui saat pemain melewati *checkpoint* selanjutnya.

- **Editable Game Object**

Level pada sebuah *campaign* dapat diubah dengan *key* yang dimiliki pemain, sehingga mengubah kondisi level tersebut. *Key* yang dimiliki pemain ini saat berada pada map, akan dianggap sebagai *object* yang dapat digunakan untuk berinteraksi. Interaksi mencakup melompat, berjalan dan *movement* lain yang dapat dilakukan pemain.

- **Multi AI NPC (Friendly)**

Terdapat NPC yang tidak akan menyerang player, NPC ini akan berupa karakter villager pada level pertama di dalam game. NPC ini dapat memiliki beberapa tipe AI. Detail tipe AI yang ada adalah sebagai berikut:

1. Villager akan menggunakan state AI wandering dan idle. Sehingga akan terlihat para villager tersebut seperti berjalan di sekeliling desa
2. Animal (Pet) akan menggunakan state AI following dan idle. Sehingga hewan seperti kucing atau anjing akan mengikuti pemain saat pemain berada di desa tersebut.

- **Multi AI NPC (Enemy)**

Musuh yang ada pada suatu level bisa memiliki tipe AI yang berbeda. Tipe AI dapat dipengaruhi oleh tipe musuh, *range* senjata musuh (apakah musuh *melee* atau *range*), kemampuan pergerakan musuh (apakah musuh dapat terbang atau hanya berjalan), ukuran *vision cone* musuh dan lain sebagainya. Berikut adalah detail dari NPC enemy yang dapat ditemukan di dalam game ini:

1. Werewolf merupakan AI enemy dengan tipe serangan melee dan tipe gerakan berjalan. Werewolf harus mendekati pemain terlebih dahulu sebelum dapat menyerang pemain. Sehingga werewolf akan menggunakan state AI patrolling, idle, following, dan attacking. Di mana following AI tersebut akan mendekati pemain hingga posisi pemain dan body AI sangat dekat atau bersentuhan, barulah AI akan berusaha menyerang pemain.
2. Ninja merupakan AI enemy dengan tipe serangan range dan tipe gerakan berjalan. Sehingga ninja akan menggunakan state AI idle, dan attacking. Di mana ninja akan ditempatkan di posisi tertentu, dan akan menyerang pemain jika berada pada area penglihatan atau line of sight dari ninja tersebut.

Karena tipe serangan ninja adalah range, maka ninja tidak perlu bergerak untuk menyerang, selama pemain masih terlihat.

3. Shaman merupakan AI enemy dengan tipe serangan melee dan tipe gerakan terbang. Shaman harus mendekati pemain terlebih dahulu sebelum dapat menyerang, namun karena shaman dapat terbang, maka shaman tidak dapat didapat terpengaruh oleh key yang menghalangi pathing seperti kedua AI sebelumnya. Sehingga shaman akan menggunakan state AI patrolling, idle, following, dan attacking.

- **AI behaviour Manipulation**

AI pada musuh juga dapat dimanipulasi oleh *action* dari player. Player dapat melakukan hal seperti menutupi *line of sight* AI, berpindah tempat dari posisi terakhir AI menangkap posisi player dan lain sebagainya. *Manipulation behaviour* ini dapat merubah kondisi AI maupun *state* yang sedang digunakan AI tersebut.

- **Save Campaign**

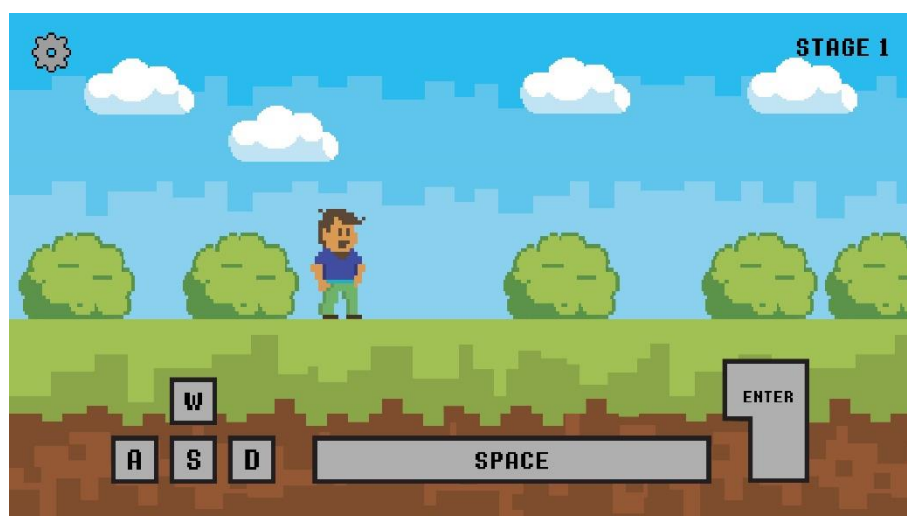
Pemain dapat menyimpan data *campaign* yang sedang dimainkan saat ini pada *in game option*. *Save campaign* secara *default* akan melakukan *overwrite save file* pada *current campaign* sebelumnya.

- **Load Campaign**

Jika terdapat data *save file*, maka player dapat melakukan *load file* untuk melanjutkan *campaign* yang disimpan pada *playtime* sebelumnya. Seluruh data player, *map*, *enemy* akan di *load* sesuai kondisi saat *save campaign* dilakukan.

4.6 Mockup Tampilan

Ini adalah beberapa gambar hasil mockup dan model representasi karakter yang akan digunakan untuk game ini.



Gambar 4.8
Mock Up Interface Game
(Sumber gambar : Dokumen pribadi)

Gambar 4.8 adalah gambar tata aturan *interface* dan tampilan game secara umum. Sehingga dapat dilihat perkiraan tinggi map, lebar dan Panjang level serta *terrain* yang ada.



Gambar 4.9
Konsep Gambar Musuh Assassin
(Sumber gambar : Dokumen pribadi)



Gambar 4.10
Konsep Gambar Musuh Shaman
(Sumber gambar : Dokumen pribadi)



Gambar 4.11
Konsep Gambar Musuh Werewolf
 (Sumber gambar : Dokumen pribadi)



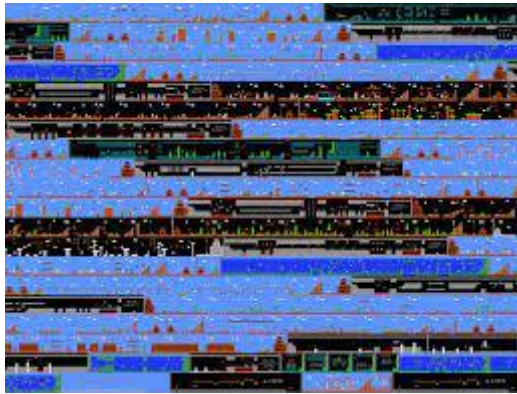
Gambar 4.9
Konsep Gambar Boss
 (Sumber gambar : Dokumen pribadi)

**Konsep gambar dan mockup dapat berubah sesuai keperluan story maupun gameplay pada hasil akhir game*

4.7 Game Referensi

Jika berbicara tentang *platformer* game, tentunya game klasik seperti Super Mario Bros menjadi salah satu game yang paling dikenang. Game ini memiliki genre *platformer action*. Super Mario Bros adalah sebuah permainan yang menggunakan karakter dengan nama Mario yang berusaha menyelamatkan seorang putri dalam sebuah kastil dengan cara mengalahkan musuh yang menghadang di jalan. Dalam perjalanan Mario dapat mengambil koin untuk point, jamur untuk tumbuh, dan jamur yang

berkedip-kedip agar Mario dapat menembak musuh tidak hanya melompat saja. Latar atau map yang digunakan dalam Game Mario Bros berupa daratan dan lorong-lorong.



Gambar 4.8
Super Mario Map

(Sumber gambar : Dokumen pribadi)

Core Gameplay berpusat pada menghindari musuh dan sistem penggerakannya adalah kunci dari *core mechanics*. Fitur lompatan dari sistem bergerak atau *movement* pada Game Super Mario memiliki peranan penting semenjak digunakan untuk menghindari musuh dan mencapai ujung dari scenario Game tersebut. Semua ini bisa saja tercakup dalam aturan desain Game untuk memberikan rasa senang bagi para pemain Game.



Gambar 4.9
Super Mario Gameplay

(Sumber gambar : Dokumen pribadi)

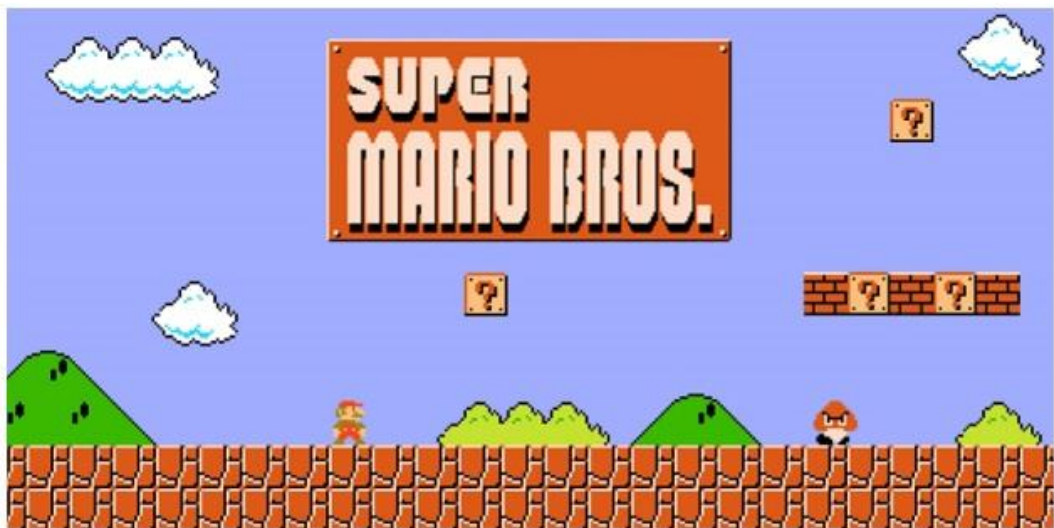
Mario harus benar-benar melompat tepat pada kepala musuh untuk bisa mengalahkan musuh tersebut. Sehingga mengalahkan semua musuh yang ada juga merupakan bagian dari *gameplay*. Berdasarkan mekanik cara player dapat mengalahkan musuh, ini hanyalah versi terselubung dari "melompat". Karenanya bisa disimpulkan bahwa dalam Game Super Mario

Bros, mengalahkan semua musuh adalah sebuah aktivitas *core meta-Gameplay* di mana semua itu bisa dilaksanakan dengan menggunakan sebuah *core mechanics* Game yang seperti melompat, dan diberikan *reward (point)* untuk melakukan *action* melompat (sebuah aktivitas *core Gameplay Mechanic*).

Pada game Super Mario Bros, musuh tidak memiliki AI, atau setidaknya, AI yang ada pada masa tersebut, sudah sangat jarang dikategorikan sebagai AI pada saat ini. Karena tingkat simplisitas dari *intelligence* yang digunakan. Musuh pada game tersebut hanya memiliki *pathing* gerak ke kanan dan kiri, di mana musuh akan bergerak ke satu arah, dan berpindah ke arah lain saat musuh menabrak object, atau masuk ke dalam jurang jika *movement* dilanjutkan. Untuk musuh yang memiliki *movement* lebih dari satu jenis (seperti boss), dilakukan *random* untuk menentukan pergerakan apa yang akan dilakukan. Mengingat hal tersebut, *behaviour* dari musuh tidak dapat diubah dan sangat mudah diprediksi.

4.8 Tabel Perbandingan

Pembuatan aplikasi game ini merupakan game yang *memiliki system gameplay* yang mirip dengan *gameplay* dengan game Super Mario Bros yang ada pada *classic console* seperti PS1, Nintendo, ataupun *Arcade Machine* kuno lainnya.



Gambar 4.11

Tampilan Super Mario Bros

(Sumber gambar : Start Screen Game Super Mario Bros)

Berikut ini adalah tabel perbandingan antara game Super Mario Bros. dengan aplikasi game yang akan dibuat :

Tabel 4.1

Tabel Perbandingan Game

No.	Fitur yang Dibandingkan	Super Mario Bros.	'Ctrl'+It
1	2D grafik dan gameplay	✓	✓
2	Singleplayer gameplay	✓	✓
3	Multiplayer Gameplay	×	×
4	Power Up	✓	×
5	AI Based Enemy	×	✓
6	Enemy Path / Patrol Route	✓	✓
7	State Machine For AI	×	✓
8	Multiple AI State	×	✓
9	Multiple AI Enemy	×	✓
10	Capable to Kill Enemy	✓	✓
11	Boss Battle	✓	✓
12	Dynamic Terrain	✓	✓
13	Multiple Level	✓	✓
14	Capable to Add Object to Level	×	✓
15	Capable to Use Object in the Level	×	✓

4.9 Uji Coba

Setelah game *platforming* ini berhasil dibuat, maka akan dilakukan uji coba dengan melibatkan minimal 30 tester. Uji coba akan dilakukan pada laptop atau komputer masing-masing user, dengan ketentuan *operating system* minimal adalah Windows 7 dan versi lebih tinggi. Dengan input berupa keyboard dan mouse.

User akan diminta menyelesaikan semua level yang diberikan. Waktu user akan dicatat dan user akan diminta mengisi sebuah *form* tentang opini user akan semua hal yang dirasakan saat bermain. Mulai dari tingkat kesulitan level, tanggapan tentang musuh, desain karakter dan *game story*. User juga akan diminta melakukan new game setelah bermain untuk menguji *overwrite save file function* yang dimiliki.

Testing juga dilakukan untuk mengecek apakah ada *bug* yang ditemukan oleh user. Jika user menemukan *bug*, maka user dapat melaporkan *bug* tersebut beserta deskripsi dan *screenshot* saat *bug* terjadi jika memungkinkan untuk membantu *debugging*. Jika *bug* memang benar ada dan dapat di rekreasi secara konsisten, maka akan dilakukan *patching* pada proses pembetulan selanjutnya untuk memperbaiki *bug* tersebut.

V. Batasan Masalah

Batasan-batasan yang ada pada perancangan game ini secara umum adalah:

1. Game dirancang untuk desktop atau laptop dengan *operating system* windows atau *operating system* lain dengan syarat memiliki C++ *visual redistributable*.

2. Game dimainkan dengan input keyboard dan mouse, dengan output ke monitor berupa tampilan game dan speaker atau *hardware* audio lainnya berupa suara *theme* dan *gameplay effect*.
3. Game dirancang dengan menggunakan grafik 2D dan 2D gameplay.
4. Tidak memerlukan koneksi internet atau menggunakan system offline.
5. Game merupakan game *singleplayer* tanpa *multiplayer* entah berupa *lan*, *coop*, maupun *internet server connected*.
6. Tidak ada fitur pembelian *item ingame* menggunakan uang sesungguhnya.

Batasan-batasan yang ada pada perancangan game secara spesifik adalah :

1. Game platformer adventure ini dapat digerakan menggunakan tombol w, a, s, d, left shift, dan spacebar. Dengan detail sebagai berikut:
 - 'w' digunakan untuk bergerak ke atas, jika terdapat interface seperti lift, tangga, atau pipa yang dapat digunakan untuk naik ke atas.
 - 'a' digunakan untuk bergerak ke kiri, jika tidak terdapat halangan, musuh ataupun border level.
 - 's' digunakan untuk bergerak ke bawah, jika terdapat interface seperti lift, tangga, atau pipa yang dapat digunakan untuk turun ke bawah.
 - 'd' digunakan untuk bergerak ke kanan, jika tidak terdapat halangan, musuh ataupun border level.
 - 'lshift' digunakan untuk menunduk, sehingga player dapat melewati tempat yang lebih rendah dari tinggi model player tersebut.
 - 'spacebar' digunakan untuk melompat, sehingga player dapat mencapai tempat yang lebih tinggi.
 - Setiap tombol tersebut akan memiliki 2d interface yang dapat digunakan sebagai object pada game.
2. Membuat game *platformer adventure* yang dapat berinteraksi dengan menggunakan tombol mouse left click dan mouse right click. Dengan detail sebagai berikut :
 - 'left click' digunakan untuk memilih dan menarik (drag) interface tombol movement untuk dijadikan object pada game. Object ini dapat digunakan untuk bersembunyi, tempat pijakan maupun penghalang bagi musuh. Selama button berada pada game sebagai object, maka pemain tidak dapat menggunakan fungsi button tersebut.
 - 'right click' digunakan untuk berinteraksi dengan object-object atau NPC pada map. Membuat AI pada *game platformer* yang memiliki beberapa *state* berupa *idle*, *patrol*, *moving*, *following*, *wandering* dan *attacking*.
3. Membuat AI pada *game platformer* yang memiliki beberapa *moving algorithm* berupa *stay*, *seek*, *arrive* dan *flee*.
4. Membuat 3 level dasar dengan detail sebagai berikut :
 - Level 1 sebagai tutorial untuk game mechanic yang akan dipakai selama game ini dimainkan.

- Level 2 yang merupakan level game untuk pengenalan semua musuh dan tipe AI yang ada di dalam game ini.
 - Level 3 yang merupakan boss fight sebagai level terakhir yang harus diselesaikan untuk menamatkan game ini.
5. Membuat setiap level memiliki tingkat kesulitan yang meningkat jika permainan dilanjutkan ke level berikutnya.
 6. Membuat level menjadi tidak dapat diselesaikan jika tidak semua *game mechanic* dipakai, namun tetap memberikan fleksibilitas bagaimana user dapat menyelesaikan gamenya.

VI. Metodologi

Metodologi yang akan digunakan untuk menyelesaikan pembuatan tugas akhir adalah metode scrum dengan langkah-langkah sebagai berikut :

1. Pembuatan dan pengajuan proposal Tugas Akhir.
2. Mencari dan mengumpulkan referensi-referensi mengenai Unity Real-Time Development Platform dan Unity Game Development Tools.
3. Studi kepustakaan tentang cara pembuatan 2D *platformer game*
4. Mencari dan mengumpulkan *sprite*, *model* dan *resource* untuk pembuatan game.
5. Membuat desain level dan *terrain* yang akan digunakan untuk game.
6. Membuat *code* dan *script* yang akan digunakan di dalam game.
7. Uji coba game pada PC dan laptop dengan Windows *Operating System*.
8. Uji coba game dengan beberapa tester yang berbeda.
9. Lakukan testing agar game berjalan dengan baik tanpa *error*.
10. Mendokumentasi hasil testing dari awal hingga akhir dalam buku laporan tugas akhir.

VII. Daftar Pustaka

1. Millington, Ian dan Funge, John. 2006. *Artificial Intelligence For Games Second Edition*. San Francisco : Morgan Kaufmann Publishers.
2. Seemann, Glenn dan Bourg, David. 2004. *AI For Game Developers*. California : O'Reilly Media.
3. Syahputra, Mohammad & Arippa, A & Rahmat, Romi & Andayani, Ulfi. (2019). *Historical Theme Game Using Finite State Machine for Actor Behaviour*. Journal of Physics: Conference Series. 1235. 012122. 10.1088/1742-6596/1235/1/012122.
4. Anonymous. 2021. "Unity Documentation (Unity Manual Version: 2020.3)". <https://docs.unity3d.com/Manual/Unity2D.html>. Diakses pada 23 Mei 2021.
5. Bevilacqua, Fernando. 24 Oct 2013. "Finite-State Machines: Theory and Implementation". <https://gamedevelopment.tutsplus.com/tutorials/finite-state-machines-theory-and-implementation--gamedev-11867>. Diakses pada 10 Juni 2021.

6. Nirwan, Debby. Januari 2012. "*Hierarchical Finite State Machine for AI Acting Engine*". <https://towardsdatascience.com/hierarchical-finite-state-machine-for-ai-acting-engine-9b24efc66f2>. Diakses pada 10 Juni 2021.
7. Anonymous. 2020. "*Mario through the years*". <https://mario.nintendo.com/history/>. Diakses pada 11 Juni 2021.