

Rancang Bangun Game 3D Edukasi Basic Web Development Menggunakan Unity 3D

Rendy Irawan¹, Yunita Sari Siregar², Mufida Khairani³

^{1,2,3} Teknik dan Komputer, Teknik Informatika, Universitas Harapan Medan, Medan, Indonesia
Email: ¹rendy9008@gmail.com, ²yunitasarisiregar1990@gmail.com, ³mufidakhairani@gmail.com

Abstrak

Perkembangan *game* menjadi media yang digemari publik dari semua kalangan yang dapat berdampak baik ataupun buruk, dimana sebuah *game* pada dasarnya dibuat sebagai media hiburan namun ada juga yang memanfaatkan *game* sebagai media edukasi. Penelitian ini dilakukan untuk membuat *game* edukasi *basic web development* yang dikemas dalam bentuk *game RPG (Role Playing Games)* dengan tujuan dapat memberikan edukasi *basic web development* yang dapat dimainkan oleh semua kalangan, dimana pembelajaran tentang *web development* saat ini lumayan banyak digemari oleh beberapa orang. Proses perancangan sistem *game* dibuat menggunakan diagram *UML* yang akan diterapkan pada aplikasi unity 3D dan penilaian yang didapat melalui kuesioner sebesar 82% responden setuju bahwa *game* ini dapat memberikan manfaat edukasi *basic web development*. *Game* ini juga di implementasikan menggunakan algoritma *finite state machine* yang mengatur perilaku musuh pada tiap *state* dan *collision detection* untuk interaksi terhadap objek yang lain. Pada proses ini menghasilkan beberapa rancangan *interface* dan juga untuk penerapan algoritma *finite state machine* diterapkan pada *animator enemy* untuk mengatur *state* perilaku *enemy*, dan algoritma *collision detection* diterapkan pada area yang memiliki *collider* untuk menampilkan *interface* dan juga memainkan *background music*.

Kata Kunci: *game, finite state machine, collision detection, unity, web development*

Abstract

The development of games is a media that is favored by the public which can have good or bad effects, where a game is basically made as for entertainment but there are also those who use games as educational media. This research was conducted to create a basic web development educational game packaged in the form of RPG (Role Playing Games) games with the result that providing basic web development education that can be played by all audience, where learning about web development is currently quite popular with some people. The game system design process is made using UML diagrams which will be applied to the unity 3D application and the assessment obtained through a questionnaire of 82% of respondents agreed that this game can be provide basic web development education. This game is also implemented using a finite state machine algorithm that controls enemy behavior in each state and collision detection for interaction with other objects. In this process create several interface designs and also for the finite state machine algorithm implemented to the enemy animator to control the state of enemy behavior, and the collision detection algorithm is applied to areas that have a collider to display the interface and also play background music.

Keywords: *game, finite state machine, collision detection, unity, web development*

1. PENDAHULUAN

Perkembangan *game* menjadi salah satu media yang saat ini banyak digemari publik dari usia muda sampai usia tua, namun perkembangan *game* bisa menjadi dampak baik maupun buruk bagi pemainnya seperti bisa memotivasi ataupun mengedukasi dan sebaliknya bisa membuat kecanduan ataupun dampak buruk lainnya. Sebuah *game* ditujukan sebagai media hiburan dan juga untuk bersaing dalam industri *game*. *Game* juga banyak digunakan sebagai media edukasi yang dikemas dalam bentuk permainan, berisi suatu materi tertentu yang digunakan untuk menunjang proses pembelajaran[1]. Maka dari itu, penelitian ini dilakukan dengan tujuan untuk membuat *game* edukasi *basic web development* yang ditujukan kepada kalangan umum, dimana pembelajaran tentang *web development* ini lumayan banyak digemari oleh beberapa orang dan penelitian ini bertujuan untuk memberikan edukasi dasar pembelajaran *web*. *Game* adalah sebuah permainan yang memiliki tujuan di setiap permainannya dan pemain dimaksudkan untuk menyelesaikan tujuan tersebut. *Game* 3D, menerapkan tiga elemen yaitu elemen x (*horizontal*), elemen y (*vertikal*), dan juga elemen z (atas-bawah), pada aturan nilai dari ketiga elemen tersebut didasari pada pengaturan kamera dalam *game* 3D, sehingga menyerupai visualisasi pada kehidupan nyata. *Game* edukasi merupakan permainan yang dibuat sebagai media hiburan dan di implementasi dengan edukasi atau pengetahuan pada konten di dalam permainannya, untuk membuat sebuah *game* memerlukan sebuah aplikasi *editor* pembuat *game* seperti unity. Unity merupakan aplikasi *editor* yang dapat digunakan untuk merancang bangun aplikasi seperti *game, augmented reality* ataupun *virtual reality*. Topik edukasi yang akan diterapkan pada konten *game* seperti edukasi *basic web development*, dan *web development* merupakan proses pembuatan sebuah *web* yang melibatkan berbagai aspek mulai dari perancangan sampai integrasi *server*.

Berdasarkan penelitian sebelumnya yaitu, pada penelitian [2] yang berjudul "*Game* Edukasi Mengenal Jajanan Tradisional Indonesia Berbasis *Role-Playing Game (RPG)*" mendapatkan kesimpulan yaitu 90% responden pada *survey* pertama dan 72% pada *survey* kedua tertarik untuk mempelajari pembelajaran memasak dari sebuah media *game* yang memperlihatkan cara memasak, dan juga beberapa resep makanan tradisional lalu, menurut [3] dalam penelitian

berjudul “*Digital Technology Implementation for Students' Involvement Base on 3D Quest Game for Career Guidance and Estimating Students' Digital Competences*”, *Game 3D* tidak hanya meningkatkan motivasi untuk belajar, tetapi juga motivasi untuk membangun sebuah karakter terhadap siswa untuk melakukan penelitian atau eksplorasi. Lalu, peneliti dari [4] dengan judul “*Virtual Environmental Enrichment through Video Games Improves Hippocampal-Associated Memory*”, menjelaskan jika stimulasi 3D sangat berdampak untuk meningkatkan daya ingat seseorang dalam otak manusia. Kesimpulan dari para peneliti tersebut para pemain memilih *game 3D* yang kompleks berdampak memiliki kemampuan untuk fokus yang lebih baik daripada orang yang fokus memainkan *game 2D*. Maka dari itu, penelitian ini dilakukan dengan tujuan untuk membuat *game* edukasi *basic web development* yang ditujukan kepada kalangan umum, dan untuk memberikan pengetahuan dasar pembelajaran *web*.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Metode yang digunakan oleh peneliti dilakukan dengan tahapan-tahapan sebagai berikut:

1. Studi Literatur
Tahap ini adalah tahap pencarian data atau informasi yang diperlukan untuk mengumpulkan data, memahami algoritma dan metode yang akan digunakan.
2. Perancangan (*Design*)
Pada tahap ini dilakukan untuk membuat rancangan aplikasi *game* dengan membuat diagram *UML* (*Unified Modelling Language*) seperti diagram *activity* dan diagram *use case*, serta membuat *design interface*.
3. Pengumpulan bahan (*Material Collecting*)
Pada tahap ini dilakukan pengumpulan asset yang sesuai dengan kebutuhan aplikasi yang di dapat pada *unity assets store*.
4. Pembuatan (*Assembly*)
Tahapan pembuatan *game* menggunakan *Unity3D* dengan mengimplementasikan rancangan dan metode *finite state machine* dan *collision detection*.
5. Pengujian Sistem (*Testing*)
Pengujian sistem dilakukan dengan hasil pembagian kuesioner, pengujian *blackbox*.
6. Distribusi (*Distribution*)
Tahap yang dilakukan setelah *game* dapat berjalan dengan baik, maka *game* akan disimpan dalam suatu media penyimpanan yang telah di *build* ke dalam sebuah *file* yang berformat *.exe* supaya dapat dimainkan pada *platform* *Windows*, atau di *publish* pada *website publish game*.

2.2 Finite State Machine

Finite State Machine (FSM) adalah sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan *state* (keadaan), *event* (kejadian) dan *action* (aksi). Proses dalam *state machine* sistem menempati satu *state* (keadaan) lalu sistem akan bertransisi menuju ke *state* lain jika mendapatkan masukan *event* tertentu dengan syarat terpenuhi nya nilai parameter dalam sebuah *event* tersebut. Setiap *state* terhubung oleh transisi dan setiap transisinya mengarah ke satu *state* lainnya. Transisi keadaan ini umumnya juga disertai oleh aksi inputan yang dilakukan oleh *system* [5].

2.3 Collision Detection

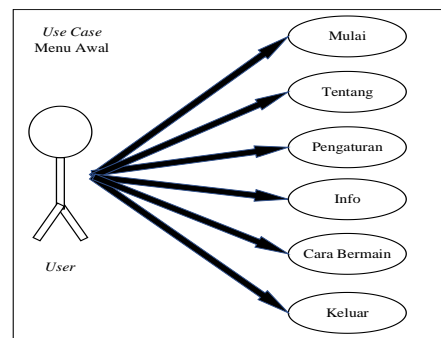
Collision Detection atau pendeteksi tumpukan adalah proses pengecekan dua atau lebih buah objek spasial tersebut saling bertumpukan atau tidak. Pada ruang spasial dua dimensi objek dikatakan saling bertumpuk apabila dua buah objek atau lebih saling beririsan. Teknik pengecekan tumpukan yang digunakan dapat dikelompokkan menjadi dua yaitu *priori detection* dan *post detection*. Pengecekan tumpukan sebelum tumpukan terjadi disebut *priori detection*, sedangkan pengecekan yang dilakukan setelah tumpukan terjadi adalah *post detection* [6].

3. HASIL DAN PEMBAHASAN

3.1 Implementasi dan Pengujian Sistem

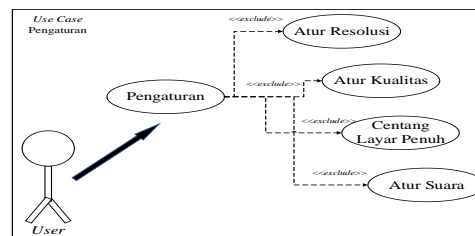
3.1.1 Usecase Diagram

Use-case diagram adalah model yang menggunakan alat pemodelan *UML* yang bersifat *open-source* yang telah dikembangkan untuk menganalisis dan mengubah *file XMI* menjadi grafik dengan mengekstraksi informasi properti dari komponen-komponen penyusun sistem, dan hubungan ataupun relasi, komponen tersebut digunakan untuk merepresentasikan interaksi antara aktor atau pengguna untuk menampilkan fungsi atau fitur yang tersedia pada sistem dan cara penggunaannya [7].



Gambar 2. Usecase Menu Awal

Keterangan pada gambar 1 *usecase* dimulai dari *user*, yang dapat memilih tombol mulai untuk memulai *game*, tombol tentang untuk melihat informasi konten dalam *game*, tombol pengaturan untuk masuk ke panel pengaturan, tombol info untuk melihat informasi pembuat *game*, tombol cara bermain untuk melihat *tutorial keybind* yang digunakan pada *game*, dan tombol keluar untuk *user* keluar dari *game*.

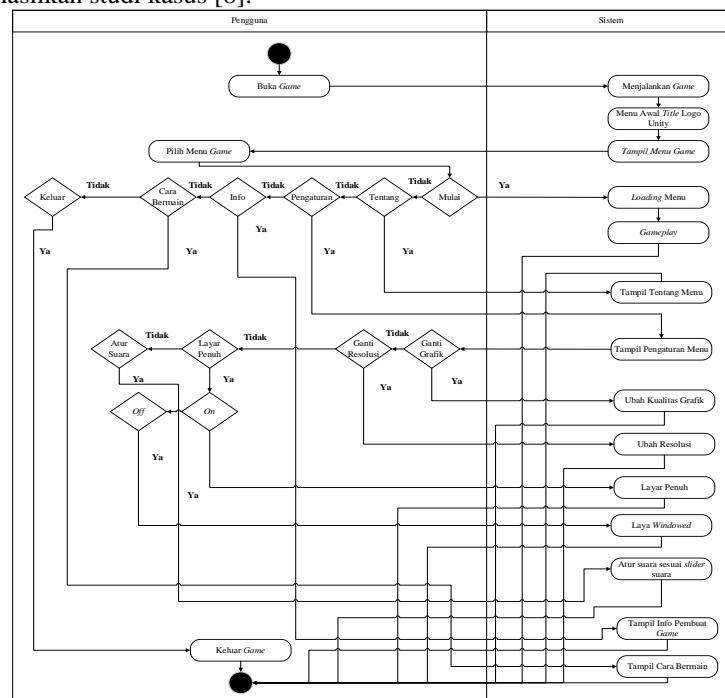


Gambar 3. Usecase Pengaturan

Keterangan pada gambar 2 *usecase* dimulai dari *user*, yang dapat mengatur resolusi *game*, mengatur kualitas dari *texture game*, mengatur tampilan *game* agar *fullscreen* atau tidak, dan juga dapat mengatur suara *game*.

3.1.2 Activity Diagram

Activity diagram adalah salah satu jenis diagram *UML* yang digunakan untuk merepresentasikan urutan tindakan atau aktivitas dalam suatu proses sistem. Namun, *Activity Diagram* memiliki sebuah simbol-simbol yang berperan penting dalam menentukan perilaku dari sistem tingkat rendah, yang dapat dimanipulasi dengan cara otomatis untuk menghasilkan studi kasus [8].

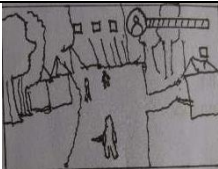

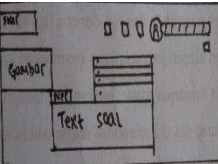
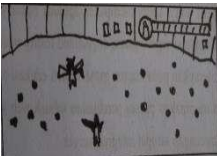


Gambar 4. Activity Diagram Game

3.1.3 Storyboard

Storyboard digunakan sebagai alat visualisasi untuk tahapan perancangan suatu *game*, yang berisi gambar atau tampilan pada *game*. *Storyboard* dapat dibuat dengan format tiga sampai empat gambar dalam satu halaman atau bisa juga satu gambar besar tiap halamannya untuk memberikan penjelasan visual lebih mendetail[9]. Visualisasi berikut menampilkan alur cerita dari *game* “*Code of Utophia*”.

Tabel 1. Storyboard

No	Gambar	Keterangan
1		Pada <i>quest</i> pertama, <i>player</i> diberikan misi untuk mengalahkan pasukan <i>bandit</i> di <i>bandit forest</i> . Pada area ini akan mengaktifkan <i>interface</i> dan <i>audio ExploringOpen.wav</i>
2		Pada <i>quest</i> kedua, <i>player</i> diberi misi untuk mengalahkan <i>monster</i> yang ada pada <i>lava mount</i> dan <i>ice mount</i> serta mengambil 2 modul yaitu modul <i>HTML</i> dan juga <i>CSS</i> yang dapat dibaca pada <i>game</i> dan pada area ini akan mengaktifkan <i>interface</i> dan juga <i>audio DangerousDungeon.wav</i>
3		Pada <i>quest</i> ketiga, <i>player</i> menjawab 20 pertanyaan <i>quiz</i> tentang <i>html</i> dan <i>css</i> agar mendapatkan senjata terkuat. Area ini akan memainkan <i>audio ChurchMedium.wav</i>
4		Pada <i>quest</i> keempat <i>player</i> harus mengalahkan empat <i>boss monster</i> . Area ini akan mengaktifkan <i>interface</i> dan <i>audio EpicBossFight.wav</i>

3.2 Implementasi Game

Implementasi *game* “*Code of Utophia*” menggunakan aplikasi *unity 3D* dan bahasa pemrograman *C#*. Tahap pengujian dilakukan saat *game* telah di *build* dan dijalankan pada *platform windows* serta hasil tabel dari kuesioner maupun hasil dari pengujian *blackbox*, yang bertujuan untuk mengetahui apakah pengujian ini telah sesuai dengan tujuan yang ingin dicapai.

3.2.1 Tampilan Game

Pada bagian ini akan menampilkan hasil dari *game* yang sudah di *build* dari *unity* ke *platform windows* sebagai aplikasi *standalone* atau berformat *exe* sebagai berikut:

1. Tampilan saat *game* dibuka

Pada tampilan ini akan menampilkan *splash image* dari logo *unity* sebagai logo pembuka ketika membuka *game*.



Gambar 5. Tampilan Splash Screen Logo Unity

2. Tampilan menu *game*

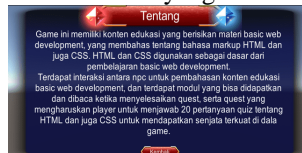
Pada tampilan ini akan menampilkan bagian panel menu *game*.



Gambar 6. Tampilan Menu Game

3. Tampilan menu tentang

Pada tampilan ini menampilkan informasi konten yang ada di dalam *game*.



Gambar 7. Tampilan Menu Tentang

4. Tampilan menu info

Pada tampilan ini menampilkan informasi dari pembuat *game*.



Gambar 8. Tampilan Menu Info

5. Tampilan menu pengaturan

Pada tampilan ini menampilkan menu pengaturan.



Gambar 9. Tampilan Menu Pengaturan

6. Tampilan menu cara bermain

Pada tampilan ini menampilkan menu cara bermain untuk informasi *keybind* yang digunakan pada *game*.



Gambar 10. Tampilan Menu Cara Bermain

7. Tampilan *loading* ketika memulai *game*

Pada tampilan ini akan muncul *bar loading* yang akan terisi penuh ketika memuat *game*.



Gambar 11. Tampilan Loading

8. Tampilan petunjuk dan pendahuluan ketika masuk ke dalam *game*

Pada tampilan ini akan menampilkan *introduction text* atau pesan dari *developer* tentang *game* kepada pemain dan petunjuk.



Gambar 12. Tampilan Pendahuluan

9. Tampilan perlengkapan

Pada tampilan ini akan menampilkan panel perlengkapan senjata jika mengklik tombol perlengkapan.



Gambar 13. Tampilan Perlengkapan

10. Tampilan peta

Pada tampilan ini akan menampilkan peta sebagai petunjuk tempat yang ada di *terrain game*.



Gambar 14. Tampilan Peta Game

11. Tampilan *quest* panel

Pada tampilan ini akan menampilkan *quest* panel jika meng klik tombol *quest*, yang berisi informasi tentang *quest* yang tersedia dan bisa diambil oleh pemain pada *village*.



Gambar 15. Tampilan Quest Panel

12. Tampilan *dialogue quest*

Pada tampilan ini akan menampilkan *dialogue npc* dari *npc quest giver* ketika *player* berinteraksi dengan salah satu *npc* pada *village*.



Gambar 16. Tampilan Dialogue Quest

13. Tampilan ketika melakukan *quest*

Pada tampilan ini merupakan tampilan ketika pemain melakukan *quest* dengan mengalahkan target sebagai objektiif dari *quest* yang telah diambil.



Gambar 17. Tampilan Melakukan Quest

14. Tampilan modul

Pada tampilan ini merupakan tampilan ketika membuka panel modul.



Gambar 18. Tampilan Modul

15. Tampilan *dialogue quiz*

Pada tampilan ini akan menampilkan panel *dialogue quiz* ketika berinteraksi dengan *npc quiz giver*.



Gambar 19. Tampilan Dialogue Quiz

16. Tampilan menyelesaikan *quiz*

Pada tampilan ini akan menampilkan panel *text* sebagai notifikasi karena menyelesaikan *quiz* dengan menjawab semua pertanyaan *quiz* dengan benar.



Gambar 20. Tampilan Notification Quiz

17. Tampilan *Game Over*

Pada tampilan ini akan menampilkan panel *game over* ketika *health point player* mencapai 0, dan pada panel ini menampilkan tombol kembali ke menu awal untuk kembali.



Gambar 21. Tampilan Game Over

18. Tampilan jeda *game*

Pada tampilan ini akan menampilkan panel jeda ketika pemain meng klik tombol *esc* pada *keyboard*, dan akan menghentikan permainan dan mematikan semua fungsi *keybind* pada *game*.



Gambar 22. Tampilan Jeda Game

19. Tampilan menyelesaikan *game*

Pada tampilan ini akan menampilkan notifikasi karena telah menyelesaikan *game* dengan mengalahkan *boss* pada *quest* ke empat.



Gambar 23. Tampilan Game Selesai

3.2.2 Pengujian *Finite State Machine* dan *Collision Detection*

Pada tahap ini akan melakukan pengujian terhadap algoritma *finite state machine* dan algoritma *collision detection* yang telah diterapkan dalam pembuatan *game*, apakah hasilnya akan sesuai dari yang di harapkan atau tidak. Proses *finite state machine* ini mencakup *action* dari *animator enemy* dan juga *animator player* yang akan di eksekusi atau dijalankan jika parameter yang ditentukan terpenuhi.



Gambar 24. FSM State Idle Enemy

Enemy berada pada *state idle* jika tidak ada perpindahan pada *state*, lalu akan berpindah *state*.



Gambar 25. FSM State Run Enemy

Enemy berada dalam *state run*, ketika *player* masuk ke jangkauan kejar, maka *enemy* akan mengejar *player*. Lalu *state* akan berpindah ke *state attack*.



Gambar 26. FSM State Attack Enemy dan Collider Attack Point

Enemy berada pada *state attack*, ketika *player* masuk ke jangkauan serang musuh, *state* ini akan berlanjut *enemy* akan melanjutkan ke *state attackB* dan *attack* selagi *player* masih masuk dalam jangkauan serang musuh. Pada tiap senjata musuh terdapat *collider attack point*, dengan menerapkan fungsi *Physics*, jadi ketika *collider* ini mengenai *object*, maka *object* akan terkena hit. Lalu akan berpindah ke *state walk*.



Gambar 27. FSM State Walk Enemy

Enemy akan berada pada *state walk*, ketika *player* berada diluar jangkauan kejar musuh selama 3 detik yang telah diatur pada *script*, maka *enemy* akan berada pada *state walk* dan kembali ke posisi awal *enemy* di tempatkan. Lalu ketika memasuki *state hurt* akan menjalankan animasi *hurt*.



Gambar 28. FSM State Hurt Enemy

Enemy berada pada *state hurt*, ketika *player* menyerang atau memberikan *damage* kepada musuh. Lalu *dead animation* akan dijalankan ketika memasuki *state dead* dan langsung memasuki *state dead end*.



Gambar 29. FSM State Dead Enemy



Gambar 30. FSM State DeadEnd Enemy

Enemy berada pada *state dead*, ketika *health point* musuh mencapai 0 dan akan berlanjut ke *state deadend*, yang akan mengubah *object* kepada *object ragdoll* dan *object* akan menghilang. Lalu, berikut ini merupakan tampilan ketika *function OnTriggerEnter* pada *collision detection* di jalankan.



Gambar 31. Audio Collider OnTriggerEnter

Player ketika masuk tiap area yang berbeda dan memasuki area *collider* akan memutar *bgm* dan menampilkan UI yang berbeda dengan menerapkan fungsi *OnTriggerEnter*, yang diatur pada *collider* yang telah diatur pada *editor*. Lalu fungsi *collider* lainnya seperti *physics raycast*.



Gambar 32. Collider Physics Raycast

Player dapat berinteraksi dengan *npc* dengan meng klik kanan pada *npc* dengan menerapkan fungsi *Physics.Raycast*, yang akan membaca jika *object* terdapat *collider* dan *tag* yang sesuai yang telah diatur pada *script*, maka akan menjalankan perintah dari *script* pada bagian ini pemain dapat memunculkan *dialogue box*.

4. KESIMPULAN

Berdasarkan hasil penelitian yang membahas mengenai "Rancangan Bangun Game 3D Edukasi Basic Web Development Menggunakan Unity 3D", maka peneliti dapat menarik kesimpulan, perancangan game "Code of Utophia" menggunakan UML diagram activity dan juga use case. Pada diagram use case dilakukan rancangan terhadap 2 sistem pada menu game dan juga sistem pengaturan yang memiliki beberapa komponen fungsi use case action. Penerapan sistem berdasarkan rancangan menghasilkan beberapa interface seperti, interface pada menu game, menu tentang, menu pengaturan, menu info, menu cara bermain, serta interface pada gameplay yang dibuat dengan menggunakan aplikasi unity dan game 3d yang dibuat berisi suatu materi edukasi basic web development yang dapat dimainkan oleh semua kalangan, dengan hasil penilaian yang didapat dari kuesioner bahwa 82% responden setuju game yang dibuat dapat memberikan edukasi basic web development kepada pemainnya. Penerapan algoritma finite state machine pada game "Code of Utophia" diterapkan pada sebuah animator enemy, yang didalam nya terdapat beberapa state dan diantara state nya ketika dijalankan object enemy akan bergerak sesuai animasi pada tiap state, setiap state dalam animator ini terdapat parameter yang kondisi nya harus terpenuhi agar state dijalankan dan kondisi tersebut diatur menggunakan script. Pada algoritma collision detection diterapkan function OnTriggerEnter, dan juga Physics Raycast, untuk function OnTriggerEnter ini diterapkan pada beberapa area terrain, yang area tersebut terdapat sebuah empty object yang memiliki collider sehingga ketika object player memasuki area tersebut function ini akan menampilkan interface dan juga memainkan audio background music yang berbeda pada setiap area terrain. Lalu, function Physics Raycast diterapkan pada object npc yang memiliki sebuah collider dan ketika player berinteraksi dengan mengklik objek tersebut maka akan memunculkan panel dialogue.

REFERENCES

- [1] E. D. Rochmada, "Pengembangan Game Edukasi Wordwall Dalam Pembelajaran IPS Materi Peninggalan Sejarah Kelas IV Sekolah Dasar," *Pgsd*, vol. 10, no. 06, pp. 1355–1364, 2022.
- [2] S. M. S. Nugroho, S. Sumpeno, and M. Liudyvia, "Game Edukasi Mengenal Jajanan Tradisional Indonesia Berbasis Role-Playing Game (RPG)," *J. Tek. ITS*, vol. 10, no. 2, pp. 450–456, 2021, doi: 10.12962/j23373539.v10i2.75320.
- [3] O. Prokhorov, V. Lisovichenko, M. Mazorchuk, and O. Kuzminska, "Digital Technology Implementation for Students' Involvement Base on 3D Quest Game for Career Guidance and Estimating Students' Digital Competences," no. January, pp. 676–690, 2022, doi: 10.5220/0010927400003364.

- [4] G. D. Clemenson and C. E. L. Stark, "Virtual environmental enrichment through video games improves hippocampal-associated memory," *J. Neurosci.*, vol. 35, no. 49, pp. 16116–16125, 2015, doi: 10.1523/JNEUROSCI.2580-15.2015.
- [5] W. Safitra, A. Faisol, and S. Adi Wibowo, "Application of the Finite State Machine Method to Non Player Character (NPC) Action Strategy Game 'Ouroboros,'" *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 4, no. 2, pp. 292–297, 2020, doi: 10.36040/jati.v4i2.2828.
- [6] F. Alamsyah, W. Diwa, and A. Yunus, "Implementasi Algoritma Collision Detection Dan Finite State Machine Untuk Karakter Musuh Pada Game Bertipe Metroidvania," *RAINSTEK J. Terap. Sains Teknol.*, vol. 1, no. 2, pp. 8–13, 2019, doi: 10.21067/jtst.v1i2.3062.
- [7] M. N. Arifin and D. Siahaan, "Structural and Semantic Similarity Measurement of UML Use Case Diagram," *Lontar Komput. J. Ilm. Teknol. Inf.*, vol. 11, no. 2, p. 88, 2020, doi: 10.24843/lkjiti.2020.v11i02.p03.
- [8] A. Jaffari, C. J. Yoo, and J. Lee, "Automatic test data generation using the activity diagram and search-based technique," *Appl. Sci.*, vol. 10, no. 10, pp. 9–13, 2020, doi: 10.3390/AP10103397.
- [9] M. I. Adithama and Ramadhan, "Pembuatan Storyboard Dalam Perancangan Animasi 3D Berjudul 'Ryan' Tentang Dampak Gaming Disorder," *J. eProceedings Art Des.*, vol. 6, no. 3, pp. 3362–3369, 2019, [Online]. Available: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/artdesign/article/view/10989/10857>
- [10] T. K. Arifonang, "Pengenalan Algoritma Pada Pembelajaran Pemrograman Komputer," *Angew. Chemie Int. Ed.* 6(11), 951–952., vol. 4, no. March, pp. 763–773, 2020.
- [11] M. Bancila, R. Rialdi, and A. Sharma, *Learn C# Programming A Guide to Building a Solid Foundation in C# Language for Writing Efficient Programs*. Packt Publishing, 2020.
- [12] S. Damarjati and A. Miatun, "Pengembangan Game Edukasi Berbasis Android sebagai Media Pembelajaran Berorientasi pada Kemampuan Berpikir Kritis," *ANARGYA J. Ilm. Pendidik. Mat.*, vol. 4, no. 2, 2021, doi: 10.24176/anargya.v4i2.6442.
- [13] A. Gani, "Microsoft Windows," *Comput. Busses*, pp. 547–568, 2020, doi: 10.1201/9781420041682.axh.
- [14] J. Hocking, *Unity in Action, Third Edition: Multiplatform Game Development in C#*, Berilustra. Simon and Schuster, 2022.
- [15] N. Khesya, "Mengenal Flowchart dan Pseudocode Dalam Algoritma dan Pemrograman," *Preprints*, vol. 1, pp. 1–15, 2021, [Online]. Available: <https://osf.io/dq45f>
- [16] H. Kurniawan, F. Syafa'at, E. Budihartono, T. Ansyor Lorosae, and D. Apriana, *BELAJAR WEB PROGRAMMING : Referensi Pengenalan Dasar Tahapan Belajar Pemrograman Web Untuk Pemula*. PT. Sonpedia Publishing Indonesia, 2023.
- [17] P. Manurung, "Multimedia Interaktif Sebagai Media Pembelajaran Pada Masa Pandemi Covid 19," *Al-Fikru J. Ilm.*, vol. 14, no. 1, pp. 1–12, 2021, doi: 10.51672/alfikru.v14i1.33.
- [18] A. Marjuni and H. Harun, "Penggunaan Multimedia Online Dalam Pembelajaran," *Idaarah J. Manaj. Pendidik.*, vol. 3, no. 2, p. 194, 2019, doi: 10.24252/idaarah.v3i2.10015.
- [19] M. Muiz and A. Arrahman, "Penerapan Collision Detection Pada Game Platformer 'Culture Seeker,'" *e-Jurnal JUSITI (Jurnal Sist. Inf. dan Teknol. Informasi)*, vol. 11, no. 01, pp. 101–111, 2022, doi: 10.36774/jusiti.v11i1.915.
- [20] W. Novayani, "Game Genre for History Education Game based on Pedagogy and Learning Content," *J. Komput. Terap.*, vol. 5, no. Vol 5 No 2 (2019), pp. 54–63, 2019, doi: 10.35143/jkt.v5i2.3360.
- [21] J. Osis and U. Donins, "Topological UML Modeling," *TopUML Model.*, no. January 2017, pp. 133–151, 2017, doi: 10.1016/b978-0-12-805476-5.00005-8.
- [22] M. Ozkaya, "Are the UML modelling tools powerful enough for practitioners? A literature review," *IET Softw.*, vol. 13, no. 5, pp. 338–354, 2019, doi: 10.1049/iet-sen.2018.5409.
- [23] M. F. Rahadian, A. Suyatno, and S. Maharani, "Penerapan Metode Finite State Machine Pada Game 'The Relationship,'" *Inform. Mulawarman J. Ilm. Ilmu Komput.*, vol. 11, no. 1, p. 14, 2016, doi: 10.30872/jim.v11i1.198.
- [24] N. F. Ramadhanti, M. Lamada, and R. Muhammad, "Pengembangan Aplikasi Game Edukasi 3D 'Finding Geometry' Berbasis Unity Sebagai Media Pembelajaran Bangun Ruang Matematika," *J. Mediat.*, vol. 4, no. 2, pp. 21–26, 2021.
- [25] A. Saputra, *Buku Sakti HTML, CSS & Javascript : Pemrograman Web Itu Gampang*. Anak Hebat Indonesia, 2019.
- [26] A. O. Sari, A. Abdillah, and Sunarti, *Web Programming*, Edisi Pert. Yogyakarta: GRAHA ILMU, 2019.
- [27] J. Simarmata, R. Aida Hanum, D. Situmorang, and M. Sitorus, *Elemen-Elemen Multimedia untuk Pembelajaran*. Yayasan Kita Menulis, 2020.
- [28] S. Sintaro, "Rancang Bangun Game Edukasi Tempat Bersejarah Di Indonesia," *J. Inform. dan Rekayasa Perangkat Lunak*, vol. 1, no. 1, pp. 51–57, 2020, doi: 10.33365/jatika.v1i1.153.
- [29] K. M. Valentino, U. Majorsy, and V. Valentine, "Permainan Q'zzie Sebagai Aplikasi Drill And Practice Menggunakan Game Engine Unity," *Ug J.*, vol. 16, pp. 32–43, 2022.
- [30] S. Widoretno, D. Setyawan, and Mukhlison, "Efektifitas Game Edukasi Sebagai Media Pembelajaran Anak," *Transform. Pembelajaran Nas.*, vol. 1, pp. 287–295, 2021, [Online]. Available: <https://ojs.uniwaru.ac.id/index.php/protrapenas/article/view/218>
- [31] I. Yuniarto and K. Adhiyarta, "Jurnal Review: Perbandingan Sistem Operasi Linux Dengan Sistem Operasi Windows," *Jupiter J. Comput. Inf. Technol.*, vol. 1, no. 1, pp. 1–7, 2021, doi: 10.53990/cist.v1i1.77.