

TUGAS AKHIR GRAFIKA KOMPUTER
“GAME DAN ANIMASI FPP 3D PAKUAN REDEMPTION”



Dosen Pengampu : Dini Suhartini, S.Kom., MMSI

Disusun Oleh :

- | | |
|-------------------------|-------------|
| 1. Muhammad Azril Hakim | (065122129) |
| 2. Revanza Permana | (065122128) |
| 3. Daud Ibrahim | (065122133) |
| 4. Ridho Dwi Nariori | (065122139) |

PROGRAM STUDI ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PAKUAN BOGOR
2024

KATA PENGANTAR

Dengan rasa syukur yang mendalam, saya menghadirkan laporan ini sebagai bagian dari penyelesaian tugas akhir mata kuliah Grafika Komputer. Proyek ini merupakan hasil jerih payah dan dedikasi kami dalam mempelajari konsep-konsep dasar Grafika Komputer serta menerapkannya dalam sebuah proyek praktis. Melalui laporan ini, kami berharap dapat menyampaikan pemahaman kami tentang topik ini serta memberikan kontribusi yang bermanfaat bagi pengembangan ilmu Grafika Komputer di masa yang akan datang.

Kami dengan senang hati menyampaikan laporan tugas akhir mata kuliah Grafika Komputer ini. Melalui penelitian ini, kami berusaha memahami konsep-konsep dasar dan menerapkannya secara praktis. Terima kasih kepada Dosen Pengampu atas bimbingan dan dukungan yang diberikan selama proses pembelajaran. Semoga laporan ini dapat bermanfaat bagi pengembangan ilmu Grafika Komputer di masa depan.

Penulis,

Bogor, 8 Juni 2024.

DAFTAR ISI

TUGAS AKHIR GRAFIKA KOMPUTER “GAME DAN ANIMASI FPP 3D PAKUAN REDEMPTION”	i
PROGRAM STUDI ILMU KOMPUTER FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM UNIVERSITAS PAKUAN BOGOR 2024	i
KATA PENGANTAR	ii
DAFTAR ISI.....	iii
BAB I PENDAHULUAN.....	1
B. Ruang Lingkup.....	2
C. Manfaat	2
BAB II TINJAUAN PUSTAKA	4
2.1. Animasi.....	4
2.2. Software Blender	4
2.3. Dimensi.....	4
2.4. Ursina Engine.....	5
BAB III PEMBAHASAN.....	6
3.1. Deskripsi Program	6
3.2. Source Code	6
BAB IV KESIMPULAN DAN SARAN	27
4.1. Kesimpulan	27
4.2. Saran	27
DAFTAR PUSTAKA	29

BAB I

PENDAHULUAN

A. Latar Belakang

Dalam era di mana permainan video telah menjadi salah satu bentuk hiburan yang paling diminati, pengembangan game dan animasi 3D menjadi semakin relevan dalam pendidikan teknologi informasi. Tugas akhir ini bertujuan untuk menyelami aspek kreatif dan teknis dari pengembangan permainan komputer, dengan fokus pada pembuatan game dan animasi 3D yang diberi judul "FPP Pakuan Redemption". Dengan menggunakan teknologi grafika komputer terkini, proyek ini akan menciptakan pengalaman interaktif yang memikat bagi pemain, sambil mengeksplorasi kemungkinan visual yang tak terbatas.

Penelitian sebelumnya telah menyoroti pentingnya integrasi yang harmonis antara gameplay yang menarik dan kualitas visual yang memukau dalam menciptakan pengalaman bermain yang memuaskan. Studi oleh Tan et al. (2017) menunjukkan bahwa kualitas grafis yang tinggi dapat meningkatkan daya tarik estetika dan imersi dalam permainan, yang pada gilirannya meningkatkan kepuasan pengguna. Dalam konteks ini, "FPP Pakuan Redemption" bertujuan untuk menggabungkan kedua aspek ini dengan baik, menciptakan lingkungan yang menarik secara visual sambil menyajikan gameplay yang menantang.

Selain itu, penelitian juga telah menyoroti peran penting animasi 3D dalam meningkatkan keterlibatan pengguna dalam lingkungan virtual. Misalnya, studi oleh Lee et al. (2019) menunjukkan bahwa animasi 3D yang realistis dapat menciptakan pengalaman yang lebih imersif dan memikat bagi pemain. Dengan demikian, dalam "FPP Pakuan Redemption", animasi 3D akan digunakan untuk memberikan detail yang kaya dan realisme yang diperlukan untuk menciptakan dunia game yang hidup dan dinamis.

B. Ruang Lingkup

Ruang lingkup pembuatan program ini mencakup dua aspek utama yaitu, pembuatan animasi mobil yang bergerak dan pembuatan objek gedung di dalam lingkungan game. Untuk animasi mobil, kami akan mengembangkan algoritma yang memungkinkan mobil bergerak dengan lancar dan responsif, serta memastikan integrasi yang tepat ke dalam lingkungan game, termasuk deteksi tabrakan dan respons terhadap interaksi pengguna, dan menggunakan tipe kamera first proper person, serta pengembangan third proper person.

Sedangkan untuk pembuatan objek gedung, fokusnya adalah pada pengembangan algoritma untuk merancang dan menggambar garis-garis yang membentuk bangunan, serta melakukan transformasi objek seperti rotasi dan translasi untuk menempatkan bangunan secara akurat di dalam lingkungan game. Selain itu, akan diimplementasikan algoritma pembuatan garis untuk membuat struktur dinding atau batasan, serta algoritma pembuatan poligon untuk menciptakan bangunan kompleks seperti gedung-gedung model MIPA, sehingga mencakup semua elemen yang diperlukan untuk membangun lingkungan game yang menarik dan responsif.

C. Tujuan

Adapun tujuan dalam pembuatan program ini adalah sebagai berikut :

- Mengembangkan game dan animasi 3D sederhana
- Mempelajari dan mengimplementasikan penggunaan kamera First Proper Person (FPP) dan Third Proper Person (TPP) dalam game.
- Menciptakan lingkungan kampus 3D yang realistis dan interaktif sebagai latar belakang permainan.
- Memenuhi tugas akhir yang diberikan oleh dosen pengampu.

D. Manfaat

Adapun sekiranya manfaat dari pembuatan program yang telah kami buat ini sedikitnya adalah sebagai berikut :

- Mengembangkan kemampuan dalam grafika komputer pemrograman game dan animasi 3D
- Memperoleh wawasan mendalam tentang proses dan dinamika industri game, dari konsep hingga peluncuran.

- Menyalurkan kreatifitas dalam merancang karakter, lingkungan , dan alur cerita dalam game dan animasinya.
- Memperkuat kemampuan bekerja sama dengan rekan tim.
- Mengembangkan keterampilan analitis dan inovatif untuk mengatasi tantangan teknis dan desain yang muncul selama pengembangan program.

BAB II

TINJAUAN PUSTAKA

2.1. Animasi

Animasi adalah teknik untuk menciptakan ilusi gerak dengan menampilkan serangkaian gambar atau frame secara berurutan dengan cepat. Setiap frame adalah gambar yang sedikit berbeda dari gambar sebelumnya, dan ketika ditampilkan secara berurutan dalam waktu singkat, mata manusia mempersepsikan perubahan tersebut sebagai gerakan mulus. Animasi dapat dibuat secara tradisional dengan menggambar setiap frame secara manual atau secara digital menggunakan perangkat lunak khusus. Teknik ini digunakan dalam berbagai media, seperti film, televisi, video game, dan iklan, untuk menceritakan cerita, menyampaikan pesan, atau sekadar menghibur.

2.2. Software Blender

Blender adalah perangkat lunak sumber terbuka (open-source) yang digunakan untuk membuat grafik 3D, animasi, model, dan rendering. Dikembangkan oleh Blender Foundation, Blender menyediakan berbagai fitur canggih yang memungkinkan pengguna untuk melakukan modeling, sculpting, texturing, animating, dan compositing. Selain itu, Blender memiliki alat untuk simulasi fisika, efek visual, pengeditan video, dan bahkan pembuatan game. Karena sifatnya yang open-source, Blender gratis untuk digunakan dan didukung oleh komunitas pengguna dan pengembang yang aktif. Dengan antarmuka yang fleksibel dan fitur yang kuat, Blender digunakan oleh profesional di berbagai industri, termasuk film, game, dan desain arsitektur, serta oleh hobiis dan pengajar.

2.3. Dimensi

Dimensi dalam grafika komputer dan game development merujuk pada jumlah koordinat yang diperlukan untuk menggambarkan posisi suatu titik dalam ruang. Dalam 2D (dua dimensi), hanya diperlukan dua koordinat, X dan Y, untuk menggambarkan objek pada permukaan datar, seperti gambar atau peta. Dalam 3D (tiga dimensi), ditambahkan koordinat Z untuk menggambarkan kedalaman, memungkinkan penciptaan objek dan lingkungan yang lebih realistis. Dimensi 4D sering kali memasukkan aspek waktu sebagai dimensi keempat, berguna dalam simulasi dan animasi yang memperhitungkan perubahan posisi sepanjang waktu.

2.4. Ursina Engine

Ursina Engine adalah framework pengembangan game open-source yang berbasis Python, dirancang untuk memudahkan pembuatan game 3D dan aplikasi grafis interaktif. Dengan sintaks yang sederhana dan intuitif, Ursina memungkinkan pengembang, termasuk pemula, untuk fokus pada logika game dan desain tanpa harus khawatir tentang detail teknis yang rumit. Engine ini menyediakan fitur lengkap seperti rendering 3D, kontrol input, pencahayaan, dan efek partikel. Dukungan komunitas yang aktif juga membantu pengembang dengan berbagai tutorial, dokumentasi, dan contoh kode, membuat pengembangan menjadi lebih cepat dan menyenangkan.

BAB III

PEMBAHASAN

3.1. Deskripsi Program

"Game dan Animasi 3D FPP Pakuan Redemption" adalah sebuah program inovatif yang menggabungkan elemen permainan dan animasi 3D dengan latar belakang kehidupan kampus Universitas Pakuan. Bercerita tentang seorang polisi yang menghadapi serangan teroris di kampus, program ini menggunakan kamera First Proper Person (FPP) untuk memberikan pengalaman yang imersif dan personal, serta mengembangkan mode Third Proper Person (TPP) untuk pandangan yang lebih luas. Fitur utama termasuk animasi 3D. Program ini tidak hanya menghibur, tetapi juga edukatif, mengajarkan pentingnya keberanian dan kesiapsiagaan di lingkungan akademis.

3.2. Source Code

```
from ursina import *

from ursina.prefabs.first_person_controller import FirstPersonController


app = Ursina()


ujungkiri = -60

ujungkanan = 60

kecepatan = 8

kecepatan2 = 18


bullets = []

moving_targets = []
```

```

for i in range(9):

    x = random.randrange(15)

    y = random.randrange(1)

    z = random.randrange(15)

    moving_target = Entity(

        model='terrorist.obj',

        color=color.white,

        scale=(1, 1, 0.1),

        dx=0.05,

        position=(x, y, z),

        rotation_y=180,

        collider='mesh')

    moving_targets.append(moving_target)


def play_sound():

    sound = Audio('suaratembakan.mp3',autoplay=True)


def musik_kemenangan():

    sound = Audio('suarakemenangan.mp3',autoplay=True)


def start_music():

    music = Audio('backsound1.mp3', autoplay=True, loop=True)

    music.volume = 0.75

start_music()

```

```

try:

    def show_image():

        image_entity.texture = load_texture('tembakan')

        image_entity.enabled = True

        time.sleep(2)

        image_entity.enabled = False


    def on_left_click():

        show_image()


    image_entity = Entity(model='quad', texture=None, scale=(1.5, 1),
enabled=False)


    if held_keys['Left down mouse']:

        on_left_click()


    print("Model tembakan berhasil dimuat.")

except Exception as e:

    print(f"Error loading gedung model: {e}")


gun = Entity(parent=camera,

    model='Pistol1.glb',

    color=color.brown,

    texture='silver',

```

```
origin_y=-0.5,  
scale= (0.2,0.5,2),  
position=(2,-1,2.5),  
collider='box')
```

```
def destroy_bullet(bullet):
```

```
    if bullet in bullets:
```

```
        bullets.remove(bullet)
```

```
        destroy(bullet)
```

```
def input(key):
```

```
    global bullets
```

```
    if key == 'left mouse down':
```

```
        bullet = Entity(  
            model='cube',
```

```
            scale=.1,
```

```
            position=(0.4, -0.4, 0.6),
```

```
            rotation_y=45,
```

```
            speed=20,
```

```
            color=color.brown,
```

```
            collider='box'
```

```
        )
```

```
        bullet.world_position = gun.world_position
```

```
        bullet.world_rotation = gun.world_rotation
```

```

        bullets.append(bullet)

        gun.blink(color.white)

        invoke(destroy_bullet, bullet, delay=3)

        play_sound()

sky = Sky(

    parent=scene,

    model='sphere',

    texture='night_sky_2',

    scale=500,

    double_sided=True

)

player_speed_normal = 4

player_speed_fast = 5

player_speed = player_speed_normal

player_start_position = Vec3(0, 5, 0)

def check_player_position():

    global player

    if player.Z < -15:

        player.position = player_start_position

```

```

try:

    gedung_model = Entity(

        model="gedung2.obj",

        texture='solid',

        scale=(0.4, 0.5, 0.6),

        position=(20, 0.01, 10),

        # collider='mesh'

    )

    print("Model gedung berhasil dimuat.")

except Exception as e:

    print(f"Error loading gedung model: {e}")

```

```

orangkampus1 = Entity (

    model="orangkampus1.obj",

    scale=(1, 1.3, 0.9),

    position=(25, 0, 10),

    rotation_y=-90,

    collider='box'

)

```

```

try:

    meja_model = Entity (

        model="mejo.glb",

```

```

        texture='solid',

        scale=(0.8, 0.4, 0.7),

        position=(45, 0, 18),

        rotation_y=180,

        collider='box'

    )

    print("Model meja berhasil dimuat.")
except Exception as e:

    print(f"Error loading pohon model: {e}")

try:

    orangkampus = Entity (

        model="orangkampus.obj",

        scale=(1, 1.3, 0.9),

        position=(33, 0, 38),

        rotation_y=180,

        collider='box'

    )

    print("Model orang berhasil dimuat.")
except Exception as e:

    print(f"Error loading pohon model: {e}")

try:

```

```

pohon_model = Entity (

    model="lowpoly_tree_sample.obj",

    texture='solid',

    scale=(0.7, 0.7, 0.7),

    position=(38, 0, 32),

    collider='box'

)

print("Model pohon 1 berhasil dimuat.")

except Exception as e:

    print(f"Error loading pohon model: {e}")

try:

    pohon2_model = Entity (

        model="pohon.glb",

        texture='solid',

        scale=(1, 2, 1),

        position=(35, 0, -22),

        collider='box'

    )

    print("Model pohon 2 berhasil dimuat.")

except Exception as e:

    print(f"Error loading pohon model: {e}")

try:

```



```

gedungrektor_model = Entity (

    model="building.glb",

    texture='solid',

    scale=(10, 10, 9),

    position=(40, 0, -37),

    rotation_y=90,

    collider='box'

)

print("Model Large Building berhasil dimuat.")

except Exception as e:

    print(f"Error loading pohon model: {e}")

try:

    mobilpolisi_model = Entity (

        model="mobil polisi.glb",

        texture='solid',

        scale=(0.7, 0.7, 0.7),

        position=(-57, -0.07, -20),

        collider='mesh'

    )

    mobilpolisi2 = duplicate(mobilpolisi_model, x=-40, z=-30)

    mobilpolisi3 = duplicate(mobilpolisi2, x=-20)

```

```

        print("Model mobil polisi berhasil dimuat.")

except Exception as e:

    print(f"Error loading pohon model: {e}")

try:

    gedung4_model = Entity (

        model="gedung4.obj",

        texture='solid',

        scale=(1.5, 2, 1.5),

        position=(50, -2, -20),

        collider='box'

    )

    print("Model Gedung 4 berhasil dimuat.")

except Exception as e:

    print(f"Error loading pohon model: {e}")

```

```

try:

    gedungrektorat_model = Entity (

        model="gedungrektorat.obj",

        texture='solid',

        scale=(7, 8, 5),

        position=(-37, 0, 26),

        rotation_y=90,

        collider='box'

```

```

    )

    print("Model Gedung rektorat berhasil dimuat.")

except Exception as e:

    print(f"Error loading pohon model: {e}")


try:

    soldiers = Entity (

        model="soldiers.obj",

        texture='none',

        scale=(0.2, 0.2, 0.2),

        position=(-28, 0, -75),

        collider='mesh',

        rotation_y=45

    )

    soldiers2 = duplicate(soldiers, x=-5, z=-68, rotation_y=0)

    soldiers3 = duplicate(soldiers, x=-20, rotation_y=0)


    print("Model police line berhasil dimuat.")

except Exception as e:

    print(f"Error loading pohon model: {e}")


try:

```

```

barrier = Entity (

    model="barrier.obj",

    texture='none',

    scale=(2, 2, 1),

    position=(7, 0, 5),

    collider='mesh',

    rotation_y=180,

)

barrier2 = duplicate(barrier, x=9, z=25)

barrier3 = duplicate(barrier,x=-10, z=11,)

barrier4= duplicate(barrier,x=-1, z=-2)

print("Model barrier berhasil dimuat.")

except Exception as e:

    print(f"Error loading barrier model: {e}")

try:

    policeline_model = Entity (

        model="policeline.obj",

        texture='none',

        scale=(0.8, 0.8, 0.8),

        position=(-28, 0, -70),

        collider='mesh'

    )

    policeline2 = duplicate(policeline_model, x=-10)

```

```

        policeline3 = duplicate(policeline2, x=8)

        print("Model police line berhasil dimuat.")

except Exception as e:

    print(f"Error loading pohon model: {e}")

try:

    lapanganbasket_model = Entity (

        model="lapanganbasket.glb",

        texture='none',

        scale=(1.5, 2, 1.2),

        position=(2, 0, 35),

        rotation_y=90,

        collider='mesh'

    )

    print("Model lapangan Basket berhasil dimuat.")

except Exception as e:

    print(f"Error loading pohon model: {e}")

try:

    Gedung2_model = Entity (

        model="gedung3.obj",

        texture='none',

        scale=(0.5, 0.5, 0.5),

        position=(-38, 0, -20),

```

```

        rotation_y=180,

        collider='mesh'

    )

    print("Model Gedung 2 berhasil dimuat.")

except Exception as e:

    print(f"Error loading pohon model: {e}")

```

```

car_model = Entity (

    model="car.obj",

    texture='none',

    scale=(0.6, 0.7, 0.5),

    position=(35, 0, -58),

    collider='None'

)

car1 = duplicate(car_model,x=39),

car2 = duplicate(car_model, x=31),

```

```

cardodge_model = Entity (

    model="cardodge.obj",

    texture='none',

    scale=(0.6, 0.7, 0.5),

    position=(28, 0, -55),

    collider='None'

)

```

```
cardodge1 = duplicate(cardodge_model,x=25),  
cardodge2 = duplicate(cardodge_model, x=22),
```

```
road = Entity (  
    model="jalanbesar.obj",  
    texture='none',  
    scale=(2.5, 1, 1),  
    position=(1, 0, -125),  
    collider='mesh'  
)
```

```
ground = Entity(  
    model='plane',  
    texture='rumput',  
    scale=(100, 1, 250),  
    texture_scale=(200, 200),  
    collider='box'  
)
```

```
billboard = Entity(  
    model="billboard.obj",  
    color=color.white,  
    scale=(0.5, 0.7, 0.3),
```

```
position=(-47, 0, -110),  
rotation_y=90,  
collider='box'  
)
```

```
billboard2 = Entity(  
    model="billboard2.obj",  
    color=color.white,  
    scale=(0.4, 0.5, 0.3),  
    position=(-47, 0, -75),  
    rotation_y=90,  
    collider='box'  
)
```

```
pohongede =Entity(  
    model="pohongede.obj",  
    color=color.white,  
    scale=(0.1, 0.2, 0.2),  
    position=(40, -3, -110),  
    rotation_y=-90,  
    collider='None'  
)
```

```
pohongede1 = duplicate(pohongede,x=30)
```



```
pohongede2 = duplicate(pohongede, x=20)
```

```
bangunanluar =Entity(  
    model="bangunanluar.obj",  
    color=color.white,  
    scale=(0.3, 0.2, 0.2),  
    position=(30, -1, -90),  
    rotation_y=-90,  
    collider='None'  
)
```

```
orangdihalte = Entity(  
    model="orangdihalte.obj",  
    color=color.white,  
    scale=(0.3, 0.2, 0.2),  
    position=(-35, 0, -110),  
    rotation_y=180,  
    collider='mesh'  
)
```

```
stop = Entity(  
    model="stop.obj",  
    color=color.white,  
    scale=(0.5, 0.7, 0.4),
```

```

        position=(-32, -2, -110),
        rotation_y=90,
        collider='mesh'
    )

```

```

mobilninut = Entity(
    model="mobilpolisi.obj",
    color=color.white,
    scale=(0.8, 1, 0.5),
    position=(-50, 0, -122),
    rotation_y=90,
    collider='box'
)

```

```

mobilkencang =Entity(
    model="mobilkencang.obj",
    color=color.white,
    scale=(0.5, 0.5, 0.4),
    position=(50, 0, -128),
    rotation_y=-90,
    collider='None'
)

```

```

wall = Entity(

```

```
    model='cube',  
    texture='wall',  
    collider='cube',  
    scale=(35, 10, 5),  
    position=(35, 5, -50),  
    color=color.gray  
)
```

```
wall5 = duplicate(wall, x=-35)
```

```
wall1 = Entity(  
    model='cube',  
    texture='wall',  
    collider='cube',  
    scale=(100, 10, 5),  
    position=(0, 5, 50),  
    color=color.gray  
)
```

```
wall3 = duplicate(wall1, rotation_y=90, x=-50, z=0)
```

```
wall4 = duplicate(wall3, x=50)
```

```
def update():
```

```
mobilkencang.x -= time.dt * kecepatan2
```

```
if mobilkencang.x < ujungkiri:
```

```
    mobilkencang.x = ujungkanan
```

```
mobilninut.x += time.dt * kecepatan
```

```
if mobilninut.x > ujungkanan:
```

```
    mobilninut.x = ujungkiri
```

```
global bullets
```

```
for m in moving_targets:
```

```
    m.x += m.dx
```

```
    if m.x > 9:
```

```
        m.x = 9
```

```
        m.dx *= -1
```

```
    if m.x < -9:
```

```
        m.x = -9
```

```
        m.dx *= -1
```

```
if len(bullets) > 0:
```

```
    for b in bullets:
```

```
        b.position += b.forward * 8
```

```
        hit_info = b.intersects()
```

```

        if hit_info.hit:

            for m in moving_targets:

                if hit_info.entity == m:

                    moving_targets.remove(m)

                    destroy(m)

                    break

            if len(moving_targets) == 0:

                message = Text(text = 'Kamu Membebaskan Universitas!',
scale=2, origin=(0,0), background=True, color=color.green)

                musik_kemenangan()

    if held_keys['escape']:

        application.quit()

player = FirstPersonController(model='cube', y=0, origin_y=-.5)

window.fullscreen = False

window.fps_counter.enabled = True

window.exit_button.visible = True

window.title = 'Ursina 3D Model Example'

app.run()

```

BAB IV

KESIMPULAN DAN SARAN

4.1. Kesimpulan

Kesimpulan dari pembuatan game dan animasi 3D FPP Pakuan Redemption untuk tugas akhir dalam proses akademik adalah bahwa proyek ini mencakup dua aspek utama: pembuatan animasi mobil yang bergerak dan pembuatan objek gedung dalam lingkungan game. Animasi mobil dirancang agar bergerak dengan lancar dan responsif, dengan integrasi yang tepat dalam lingkungan game, termasuk deteksi tabrakan dan interaksi pengguna, serta penggunaan kamera first person dan third person. Untuk objek gedung, fokusnya pada pengembangan algoritma untuk menggambar garis dan poligon yang membentuk bangunan, serta transformasi objek untuk penempatan yang akurat dalam game. Dengan pendekatan ini, proyek ini bertujuan menciptakan lingkungan game yang menarik dan responsif, menggabungkan animasi dinamis dan desain arsitektural yang kompleks.

4.2. Saran

Untuk pengembangan lebih lanjut dari program yang telah dibuat, beberapa saran yang dapat dipertimbangkan adalah :

- Optimasi Performa : melakukan optimasi algoritma dan penggunaan sumber daya untuk memastikan performa animasi dan game yang stabil dan responsif pada berbagai perangkat.
- Pengujian dan validasi : Melakukan pengujian ekstensif untuk mendeteksi dan memperbaiki bug dan semua kekurangan serta validasi terhadap berbagai skenario pengguna untuk meningkatkan pengalaman bermain,
- Pengembangan fitur tambahan : Tambahkan fitur tambahan seperti efek suara, soundtrack, dan elemen interaktif lainnya untuk meningkatkan imersi dan daya tarik game.
- Peningkatan visual dan Animasi : Tingkatkan kualitas visual dan animasi dengan menggunakan teknik rendering yang lebih canggih dan detail tekstur yang lebih tinggi.

- Feedback Pengguna : Libatkan pengguna dalam tahap pengujian beta untuk mendapatkan masukan berharga dan melakukan penyesuaian berdasarkan feedback mereka.

DAFTAR PUSTAKA

- Tan, C. T., Teo, J. T., & Yeo, H. L. (2017). Integrating Gameplay and Graphics for an Immersive 3D Serious Game. In International Conference on Universal Access in Human-Computer Interaction (pp. 463-474). Springer, Cham.
- Lee, J. Y., Lee, K. J., Park, S., & Lee, J. (2019). Research on Realistic 3D Animation Effects in Virtual Reality Based on Human-Computer Interaction. In Proceedings of the International Conference on Computer Graphics Theory and Applications (pp. 342-348). Scitepress.
- ChatGPT. (2024, 9 Juni). <https://chatgpt.com/c/fdb4620c-bd54-4c5e-9f5f-6cb435a16253>