

V-Fighter: Game Platformer 2D Protokol Kesehatan Dengan Menggunakan Godot (STRATEGI)

Vanessa Metayani^{#1}, Robby Tan^{#2}

[#]Program Studi SI Teknik Informatika, Universitas Kristen Maranatha
Jl. Prof. Drg. Surya Sumantri No.65, Bandung, Indonesia

¹1872044@maranatha.ac.id

²robbly.tan@it.maranatha.edu

Abstract — Covid-19 pandemic which has not ended makes public still need to comply health protocols. However, the low level of compliance with health protocols makes it difficult to break the chain of the spread of the covid-19 virus. The covid-19 virus is easier to spread to children, so it is important to teach health protocols to children. Therefore, medium for delivering health protocols is needed, especially for children aged 5-10 years old. 2D platformer video game still has many fans, so it can be one of developed alternative medium. Godot game engine is used to develop health protocols education themed 2D platformer video game. The data sources come from the internet and scientific references related to research. Game testing is executed using the user acceptance test method, namely alpha testing to test the game before it is distributed and beta testing to test the game directly on the target. The research technique used is in form of data collection from the game test result questionnaire. The game test results shows that this game can be used as medium for delivering health protocols, and the results from bug/error testing are that 66,7% of bugs/errors are found, and 33,3% of bugs/errors are not found. From the test results, can be concluded that Godot software can be used to create 2D platformer game which can provide education about the importance of health protocols for children.

Keywords— 2D platformer, covid-19, Godot, health protocols, video game.

I. PENDAHULUAN

Pandemi covid-19 dimulai sejak 2019 akhir dan belum usai. Walaupun kini status pandemi covid-19 sudah berubah menjadi endemi, masyarakat masih perlu mematuhi protokol kesehatan. Namun, berdasarkan data dari situs web covid19.go.id, terdapat 20 daerah yang kepatuhan protokol kesehatannya kurang dari 60% [1]. Dengan banyaknya ketidakpatuhan tersebut, maka sulit untuk memutus rantai penyebaran virus covid-19. Virus covid-19 dapat lebih mudah menyebar pada anak-anak, sehingga masyarakat diharapkan dapat tetap menerapkan protokol kesehatan dan ikut mengajarkan pentingnya protokol kesehatan pada anak-anak.

Edukasi mengenai protokol kesehatan dapat diajarkan pada anak-anak, mulai dari usia 5-10 tahun. Dalam rentang usia tersebut, anak-anak cenderung senang bermain. Selain itu, adanya perkembangan teknologi komputer membuat permainan video berkembang menjadi lebih modern dan menarik. Hal tersebut membuat permainan video dapat menjadi salah satu media penyampaian edukasi protokol kesehatan.

Permainan video berjenis *platformer* atau dikenal sebagai *jump and run games* merupakan jenis permainan yang mengharuskan pemain menggerakkan karakter dari 1 titik ke titik lain di arena. Pada umumnya, permainan video *platformer* memiliki tampilan visual 2D menyamping (*platformer 2D*). Hingga sekarang, masih banyak yang gemar memainkan permainan video *platformer 2D*. Godot merupakan salah satu perangkat lunak yang mendukung pembuatan permainan video *platformer 2D*.

Berdasarkan permasalahan di atas, muncul sebuah ide untuk membuat permainan video *platformer 2D* bertema edukasi protokol kesehatan bagi anak-anak menggunakan perangkat lunak Godot.

II. KAJIAN LITERATUR

A. Video Game

Video game dapat didefinisikan sebagai sebuah permainan yang dimainkan dengan peralatan audiovisual dan dapat didasarkan pada sebuah cerita fiksi. *Video game* telah beradaptasi menjadi budaya, bentuk seni, narasi, alat pendidikan, dan lain sebagainya [2]. *Video game* merupakan perkembangan dari sebuah permainan yang bisa dimainkan melalui mesin komputer, konsol, maupun ponsel. *Video game* juga memiliki banyak genre, salah satunya yang terkenal, klasik dan umum adalah *platformer 2D*.

Dalam penelitian yang berjudul “FoodKnight: A Mobile Educational Game and Analyses of Obesity Awareness In Children”, karya Thomas Bailey, Fadi Thabtah, Marcus Wright, dan Duy Ahn Tran dari Manukau Institute of Technology, New Zealand, penggunaan permainan jenis *platformer* diterapkan untuk edukasi obesitas pada anak-anak dengan gambaran karakternya menggunakan makanan cepat saji dan minuman berkarbonat sebagai musuh, sayur dan buah sebagai poin yang dikumpulkan dan karakter berbentuk bintang sebagai karakter pemain [3]. Penelitian ini menggunakan perangkat lunak Unity dan dirilis untuk android di Google Play Store. Hasil dari penelitian ini adalah *video game* dapat menjadi media belajar bagi anak-anak dan dengan adanya permainan FoodKnight, diharapkan dapat membantu mengubah kebiasaan anak-anak dari makan-makanan yang tidak sehat menjadi makan-makanan sehat sehingga tercapai pola hidup sehat dan anak-anak terhindar dari penyakit diabetes.

Pada penelitian berjudul “Flat Design Sebagai Gaya Visual Asset Game Attack of The Krona” karya I Wayan Adi Putra Yasa, dkk. dari Program Studi Manajemen, Fakultas Ekonomi, Universitas Mahasaraswati, penelitian permainan *platformer* ini menggunakan konsep *flat design* yang menonjolkan aspek pendekatan karakter yang *minimalism* (kesederhanaan), *usefulness* (kegunaan), dan *cleanliness* (kerapian). Konsep *flat design* ini memiliki prinsip penyederhanaan bentuk (minimalis) dari bentuk aslinya dan berfokus pada penggunaan warna yang solid objek tanpa tekstur, tipografi, dan UI yang lebih sederhana [4]. Dari penelitian tersebut, dapat disimpulkan bahwa *flat design* diharapkan mampu menyampaikan pesan dalam *game* agar tepat sasaran.

B. Platformer 2D

Permainan jenis *platformer* atau sering disebut *jump and run games* merupakan *subgenre* dari *action game* di mana pemain harus menggerakkan karakter dari 1 titik ke titik lain di arena. Permainan jenis ini terkenal pada era 90-an dan masih banyak yang gemar memainkannya hingga sekarang. Jenis permainan ini dicirikan dengan medan yang tidak rata dan *platform-platform* dengan ketinggian berbeda sehingga bergantung pada kemampuan karakter melompat atau memanjat untuk mencapai titik tujuan [5].

Kemampuan atau gerakan-gerakan lain juga dapat menjadi faktor penting dalam permainan seperti memanjat, berayun menggunakan tali, melompat dari satu tembok ke tembok yang lain, meluncur, melompat tinggi di udara, dan memantul dapat ditemukan dalam permainan berjenis *platformer* [5]. Permainan jenis ini biasanya sangat dinamis, menuntut kecepatan dan ketepatan pemain dalam menggunakan kontrol. Gerakan paling umum adalah berjalan, berlari, melompat, menyerang, dan memanjat. Melompat merupakan elemen gerakan yang penting dalam permainan *platformer* karena dalam permainan sering terdapat zona-zona dan musuh berbahaya yang harus dihindari.

Permainan *platformer* juga cocok untuk PC maupun *mobile* karena kesederhanaan konsepnya. Pemain hanya perlu menggerakkan karakter dari satu *platform* ke *platform* lain untuk mencapai tujuan dan menghindari rintangan yang ada karena jika gagal maka pemain harus mengulang dari awal. Permainan *platformer* biasanya memiliki tampilan visual 2D menyamping, namun perkembangan jenis *platformer* ini sudah ada yang memiliki tampilan 3D dengan tampilan sudut pandang kamera di belakang karakter atau isometrik [5]. Saat ini, ada banyak perangkat lunak yang mendukung pembuatan permainan 2D dan 3D, salah satunya adalah Godot.

C. Godot

Godot adalah mesin permainan 2D dan 3D serbaguna yang dirancang untuk mendukung semua jenis proyek. Godot dapat digunakan untuk membuat aplikasi permainan atau aplikasi yang kemudian dapat dirilis di desktop, seluler maupun web [6]. Dalam membuat permainan pada aplikasi godot, terdapat istilah-istilah yang umum dijumpai seperti *scene*, *kinematic body*, *sprite*, *tile map*, dan lain sebagainya.

1) Scene

Scene dalam godot merupakan tempat sekumpulan *node* yang disusun secara bertingkat seperti pohon [7]. Terdapat 2 jenis *scene* dalam godot, yaitu *2D scene* dan *3D scene*. Apabila permainan yang ingin dibuat menggunakan gambar 2 dimensi, maka dapat menggunakan *2D scene*. Jika permainan yang dibuat menggunakan 3 dimensi, maka dapat menggunakan *3D scene*. Setiap *scene* tersebut hanya dapat memiliki satu *root node* (*node* utama) dan banyak *sub-node* (*child node*). *Root node* dapat diganti tipenya, contohnya jika memilih *2D scene* pada saat membuat *scene* baru maka *node*

yang pertama kali muncul adalah *node2D*. *Node2D* tersebut dapat diganti dengan tipe lain salah satu contohnya *kinematic body*.

2) Kinematic Body

Kinematic body adalah sebuah rangka gerak fisik yang dapat mendeteksi tumbukan namun tidak terpengaruh gravitasi. *Kinematic body* digunakan untuk menggerakkan suatu objek yang gerakannya dapat diatur pengguna melalui kode [8]. Terdapat 2 jenis *kinematic body*, yaitu *kinematic body 2D* dan *kinematic body 3D* yang penggunaannya tergantung pada jenis *scene* yang digunakan. *Kinematic body* dapat digunakan jika memiliki bentuk tumbukan yang dikenal dengan nama *collision shape*. *Collision shape* ada 2 jenis yaitu *collision shape 2D* dan *collision shape 3D* tergantung pada *kinematic body* yang dipakai. *Collision shape 2D* memiliki banyak bentuk di antaranya *capsule shape*, *circle shape*, *concave polygon shape*, *convex polygon shape*, *line shape*, *ray shape*, *rectangle shape*, dan *segment shape*. *Kinematic body* dan *collision shape* tersebut dapat dipasangkan pada *sprite* untuk memberikan bentuk fisik pada karakter sehingga pada saat bersentuhan dengan objek tertentu, *sprite* dan objek tidak akan saling menembus.

3) Sprite

Sprite terbagi menjadi 2 jenis, yaitu *2D sprite* dan *3D sprite*. Dalam *scene 2D*, *sprite* menampilkan tekstur gambar 2 dimensi atau model 3 dimensi dalam *scene 3D* [9]. Untuk membuat *sprite* menjadi hidup atau bergerak, salah satu *node* yang dapat digunakan adalah *animation player*. Dalam *animation player*, setiap gerakan *sprite* direkam ke dalam sebuah *key frame* yang kemudian akan disusun menjadi 1 rangkaian animasi. Saat ini sudah terdapat jenis *node sprite* yang dapat sekaligus memainkan animasi, yaitu *animation sprite*. Bedanya adalah saat memasukkan tekstur gambar sudah langsung mengatur *frame* gambar secara otomatis sehingga tidak perlu memasukkan *key frame* secara manual.

4) Tile Map

Tile map merupakan petak-petak lantai yang digunakan untuk membuat tata letak permainan [10]. Terdapat 3 pilihan yang dapat digunakan dalam membuat *tile map*, di antaranya: *single tile*, *auto tile*, dan *atlas tile*. *Single tile* digunakan untuk mengambil satu petak lantai. *Auto tile* digunakan untuk mengelompokkan lantai dan mengatur lantai yang akan digambar sesuai konten sel yang berdekatan. *Atlas tile* dapat mengelompokkan lantai sekaligus dan memungkinkan membuat lantai secara acak dari grup.

III. ANALISIS DAN RANCANGAN

A. Rancangan Permainan

Pada bagian rancangan permainan ini, akan menjelaskan tahapan-tahapan dan deskripsi permainan yang akan dikembangkan.

1) Deskripsi Permainan

Permainan “V-Fighter” merupakan permainan berjenis *platformer 2D*. Pemain dapat mengumpulkan score sebanyak-banyaknya dengan mengumpulkan *hand sanitizer* yang tersebar dalam arena atau dengan menginjak musuh, kemudian berpindah arena melalui vaksin sebagai portal.

2) Player

Player adalah karakter dalam permainan yang dimainkan oleh pemain. Permainan ini dapat dimainkan oleh 1 pemain. Tipe permainan ini adalah *single player platformer game* di mana pemain bertujuan untuk mendapatkan *hand sanitizer* sebanyak mungkin tanpa menabrak musuh dan mendapatkan skor setinggi-tingginya.

3) Objective (Tujuan permainan)

Tujuan permainan ini adalah pemain bergerak dari satu titik ke titik lainnya menjelajah dan mengumpulkan collectible item (*hand sanitizer*), menghindari musuh dan mendapatkan skor yang tinggi.

4) Rules (Aturan)

- Karakter pemain awalnya diletakkan di suatu titik dalam *scene* permainan awal yang tidak ada karakter musuh.
- Pemain berusaha mengambil collectible item berbentuk *hand sanitizer* sambil menghindari musuh atau menginjaknya agar pemain mendapatkan skor setinggi mungkin hingga permainan berakhir.
- Pemain berusaha mengumpulkan *hand sanitizer* sebanyak mungkin dan selesai jika pemain gugur ketika bersentuhan dengan musuh yang bergerak.

5) Sumber Daya

Jenis sumber daya yang ada adalah beberapa collectible item berupa hand sanitizer yang berada di sekitar arena permainan.

6) Game State & Screen Flow

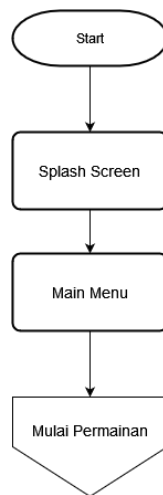
Game state dalam permainan ini merupakan suatu keadaan dalam permainan pada waktu tertentu. Dalam permainan ini terdapat 1 game state, yaitu:

- Game state collecting item merupakan game state di mana pemain berusaha mengumpulkan collectible item berupa hand sanitizer sebanyak mungkin sambil menghindari musuh atau halangan yang ada.

Game Screen Flow akan digunakan dalam menjelaskan urutan-urutan aktivitas yang terdapat dalam permainan. Game Screen Flow menggambarkan di mana alur dimulai dan akan menjelaskan detail proses dari semua event yang terdapat dalam permainan. Berikut beberapa game screen flow dalam permainan ini:

- Game Screen Flow Main Menu:

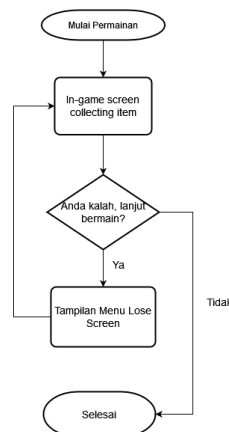
Saat pemain pertama kali menjalankan permainan akan tampak splash screen dan setelahnya akan masuk ke main menu dari permainan. Dari main menu tersebut, terdapat 2 menu: mulai permainan dan keluar. Pemain dapat melanjutkan ke dalam permainan dengan cara mengklik menu mulai permainan seperti yang tergambar dalam Gambar 1.



Gambar 1. Game Main Menu Screen Flow

- Game Screen Flow Collecting Item:

Ketika pemain memasuki tampilan permainan utama, pemain harus berusaha mencapai skor setinggi-tingginya dengan mengumpulkan collectible item hingga karakter kalah jika menabrak musuh dalam arena tersebut. Pemain dapat memilih untuk mengulang bermain permainan lagi. Alur dapat dilihat seperti pada Gambar 2.



Gambar 2. Game Screen Flow

7) Informasi

Informasi yang terdapat dalam menu instruksi pada permainan ini menginformasikan bagaimana cara memainkan permainan ini berupa teks dan gambar penjelasan cara bermain dengan disisipkan instruksi protokol kesehatan. Semua informasi mengenai sistem permainan tampak pada bagian instruksi.

8) Interaksi Pemain

Tidak terdapat interaksi antar pemain dalam permainan ini karena permainan ini merupakan permainan single player.

9) Tema

Perancangan tema dalam permainan ini menggunakan desain *pixel art*. *Pixel art* adalah karya seni digital yang diciptakan oleh lewat perangkat lunak di mana gambar dibuat dalam tingkatan pixel (elemen terkecil dalam sebuah gambar yang ditampilkan dalam layar komputer).

B. Rancangan Karakter dan Animasi

Dalam penelitian ini, karakter utama permainan adalah anak-anak dan karakter musuh adalah virus dan kerumunan. Ilustrasi karakter utama dan musuh dibuat dengan menggunakan perangkat lunak CLIP STUDIO PAINT. Penggambaran karakter dan animasi untuk karakter utama menggunakan referensi gambar zombie yang bersumber dari internet seperti pada Gambar 3.

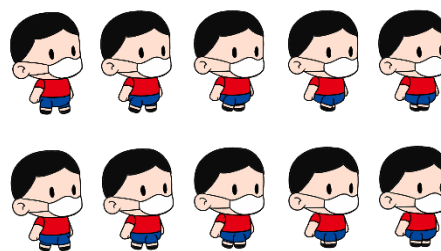


Gambar 3. Referensi Karakter dan Animasi Zombie berjalan

Berdasarkan referensi, gaya gambar yang digunakan terkesan imut meskipun sebenarnya zombie berunsur gelap. Gaya gambar imut tersebut cocok dengan kesan anak-anak. Gaya gambar tersebut adalah *chibi style* yang berasal dari bahasa Jepang (ちび = chibi) yang berarti anak kecil. Gaya gambar ini digunakan untuk menggambarkan anak-anak dalam versi karakter [11]. Tokoh karakter anak-anak yang digambarkan dalam penelitian ini memiliki memiliki 2 buah animasi, yaitu diam dan berjalan terlihat seperti pada Gambar 4 dan Gambar 5.

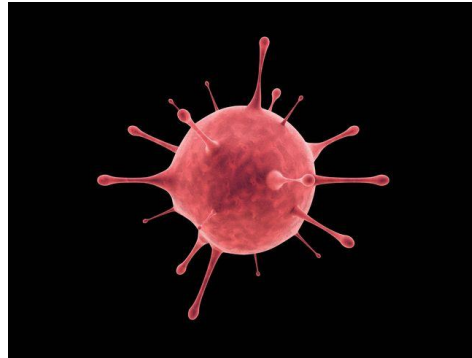


Gambar 4. Karakter dan Animasi diam yang digunakan dalam permainan



Gambar 5. Karakter dan Animasi berjalan yang digunakan dalam permainan

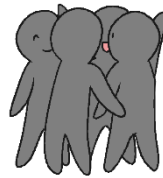
Setelah karakter utama selesai, berikutnya membuat karakter musuh, yaitu virus dan kerumunan. Karakter virus menggunakan referensi gambar yang dibuat oleh wirestock dari situs Freepik.com (Gambar 6) sedangkan untuk kerumunan tidak menggunakan referensi. Karakter dibuat menggunakan aplikasi yang sama yaitu CLIP STUDIO PAINT, namun tanpa animasi. Karakter virus (Gambar 7) dengan warna merah dan kerumunan (Gambar 8) digambarkan dalam bentuk siluet kartun manusia dengan warna abu-abu.



Gambar 6. Referensi gambar virus



Gambar 7. Karakter Virus



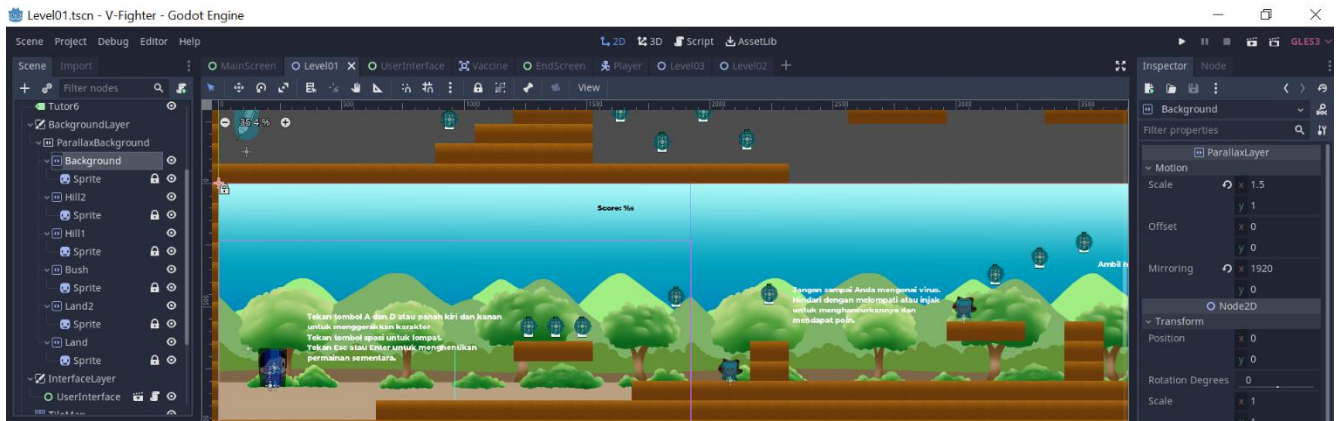
Gambar 8. Karakter Kerumunan

C. Sistem Data Pemain

Sistem data pemain disimpan ke dalam sebuah *script* dengan nama `PlayerData.gd`. Kemudian, *script* tersebut dimasukkan ke sebuah menu `AutoLoad`. Tujuannya adalah agar fungsi-fungsi yang terdapat dalam *script* dapat digunakan secara global oleh *script* lain yang menggunakan fungsi-fungsi pada `PlayerData.gd`.

D. Rancangan Latar Belakang

Latar belakang dalam permainan ini menggunakan *parallax background* yang berfungsi untuk membuat latar belakang bergerak sesuai dengan pergerakan kamera. Gambar latar belakang setiap level permainan dibuat dengan aplikasi `CLIP STUDIO PAINT`. Level 01 dengan latar belakang taman, level 02 dengan latar belakang pinggiran kota dan level 03 dengan latar belakang perumahan. Latar belakang tersebut dipilih karena merupakan tempat biasa banyak ditemukan anak-anak bermain. Untuk membuat latar belakang *parallax*, masing-masing gambar dibuat terpisah (memiliki layer). Contohnya untuk latar belakang taman, gambar tanah, pohon dan semak yang jelas berada di bagian layer pertama. Pada layer berikutnya gambar semak tinggi, layer berikutnya gambar gunung, terpisah demikian seterusnya seperti pada Gambar 9.



Gambar 9. Hasil Parallax Background

E. Rancangan Perpindahan Adegan

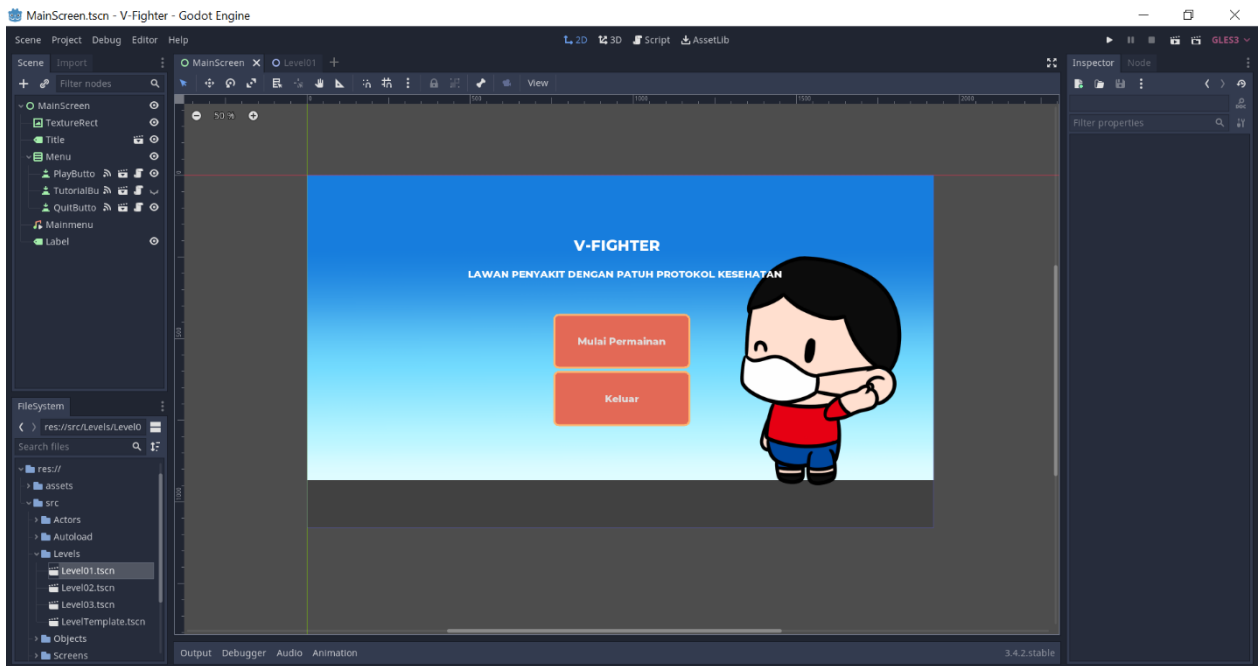
Perpindahan adegan menjadi salah satu bagian yang penting. Dalam penelitian ini, perpindahan adegan digambarkan dengan bentuk suntikan vaksin. Vaksin ini terdapat pada masing-masing level yang menggambarkan 3 kali vaksin, yaitu vaksin dosis I, dosis II, dan dosis *booster*. Gambar 10 tidak menggunakan referensi dan dibuat dengan aplikasi CLIP STUDIO PAINT. Untuk perpindahan adegan menggunakan scene yang berisi *sprite* Gambar 10 dan menggunakan script yang kemudian akan melakukan *load path scene* yang dituju.



Gambar 10. Suntikan vaksin

F. Rancangan Tampilan

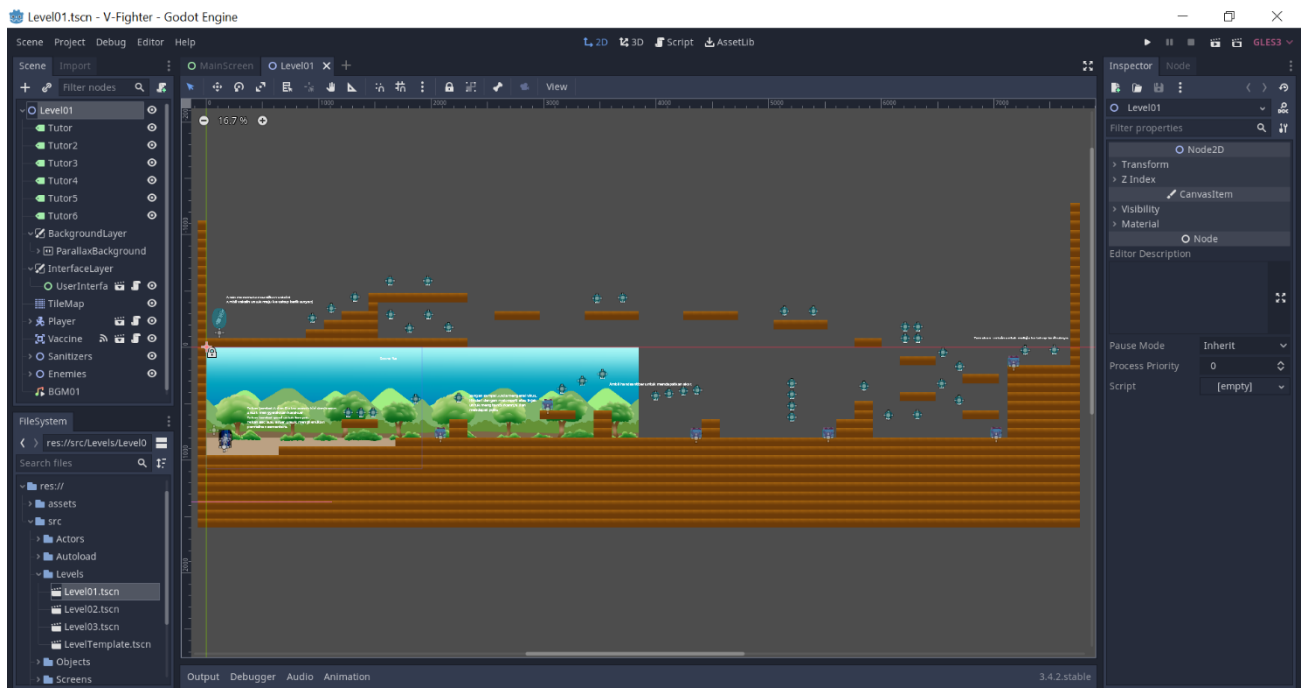
Dalam penelitian ini, pada bagian menu utama terdapat menu: mulai permainan dan keluar. Saat tombol menu mulai permainan ditekan, maka akan berpindah ke tampilan permainan dan jika tombol menu keluar ditekan, maka permainan berhenti. Rancangan menu utama dapat dilihat pada Gambar 11.



Gambar 11. Rancangan Menu Utama

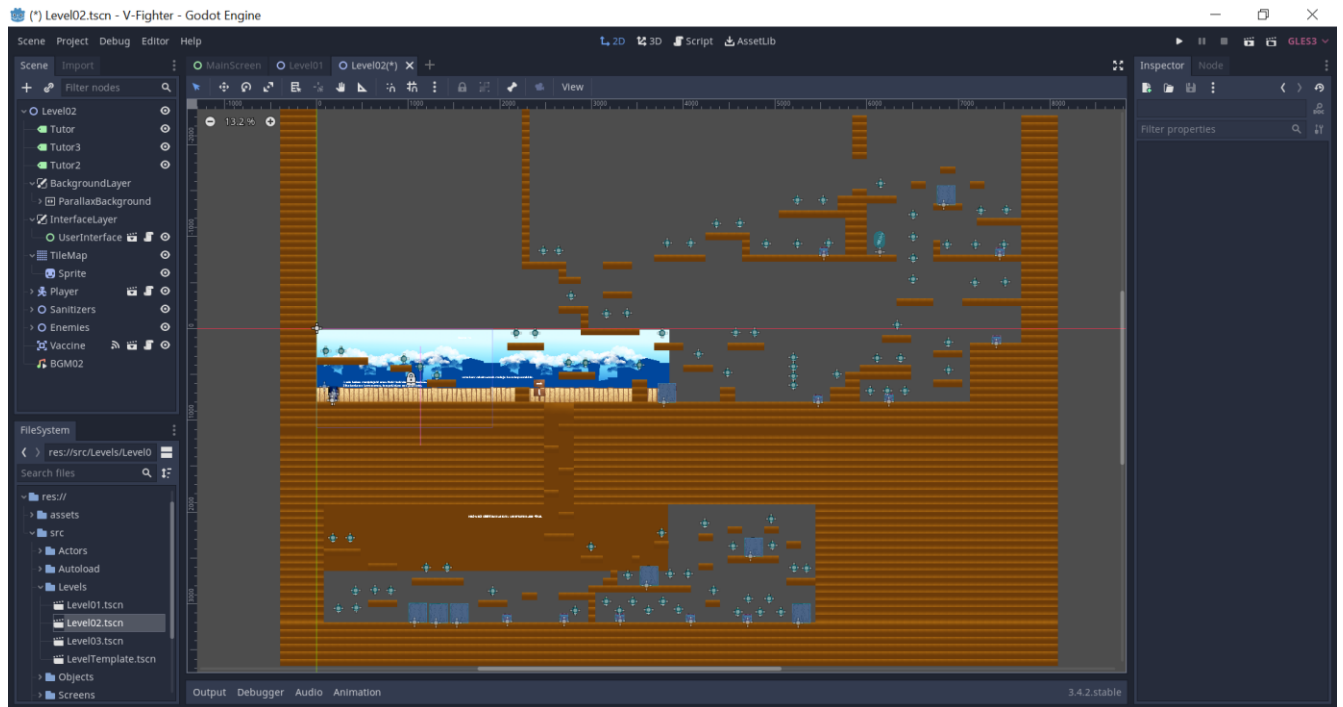
G. Rancangan Level Permainan

Terdapat 3 level dalam permainan, mulai dari level 01 hingga level 03. Masing-masing level memiliki arahan yang disisipkan dalam bentuk label di berbagai tempat di arena permainan. Level 01 pada Gambar 12 merupakan level paling mudah karena memiliki jumlah musuh yang tidak banyak. Level 01 memberikan arahan sistem kerja permainan, mulai dari tombol untuk kendali karakter bergerak dan melompat, mengambil collectible item agar mendapat skor, menghindari virus tanpa mendapat skor atau menginjak virus untuk mendapat skor, dan pada setiap level terdapat arahan untuk menemukan vaksin sebagai jalan keluar menuju ke level berikutnya. Inti dari level 01 adalah pengenalan sistem kerja permainan.



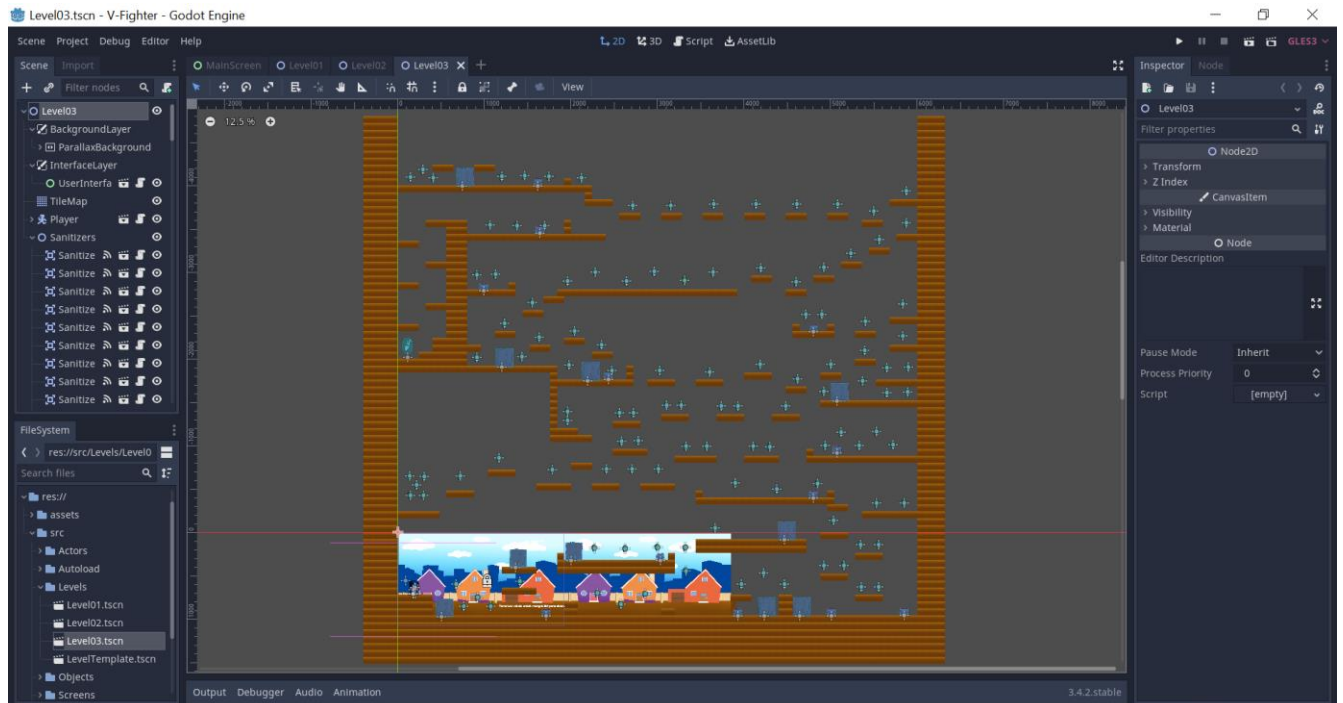
Gambar 12. Rancangan Level 01

Level 02 pada Gambar 13 selain virus, muncul juga kerumunan sebagai musuh baru yang harus dihindari pemain dengan mencari jalan lain. Dalam level 02 juga terdapat arahan untuk berhati-hati dengan kerumunan. Inti dari level 02 ini adalah agar pemain menghindari kerumunan.



Gambar 13. Rancangan Level 02

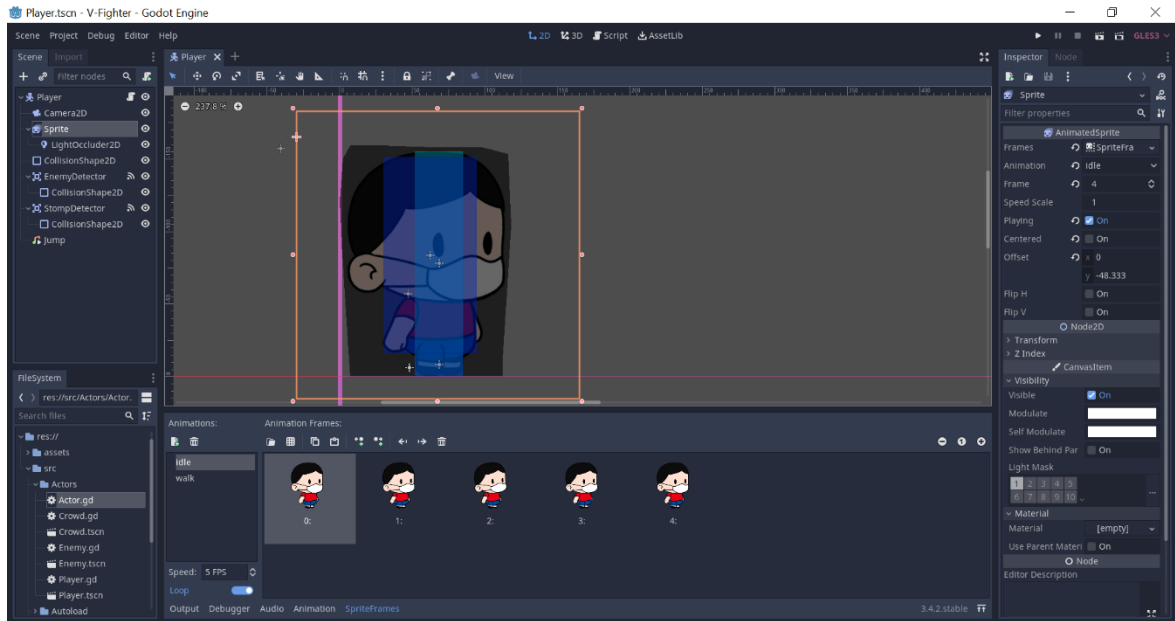
Level 03 dalam Gambar 3.17 merupakan level terakhir. Pemain harus menemukan vaksin terakhir untuk mengakhiri permainan. Inti dari level 03 adalah menyelesaikan permainan dengan menerapkan sistem kerja permainan dari level 01 dan level 02.



Gambar 14. Rancangan Level 03

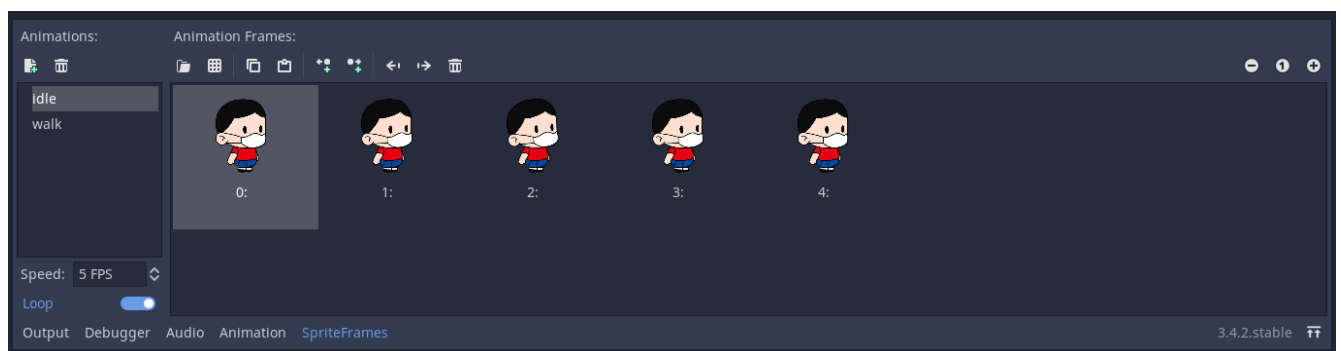
IV. IMPLEMENTASI

A. Implementasi Karakter

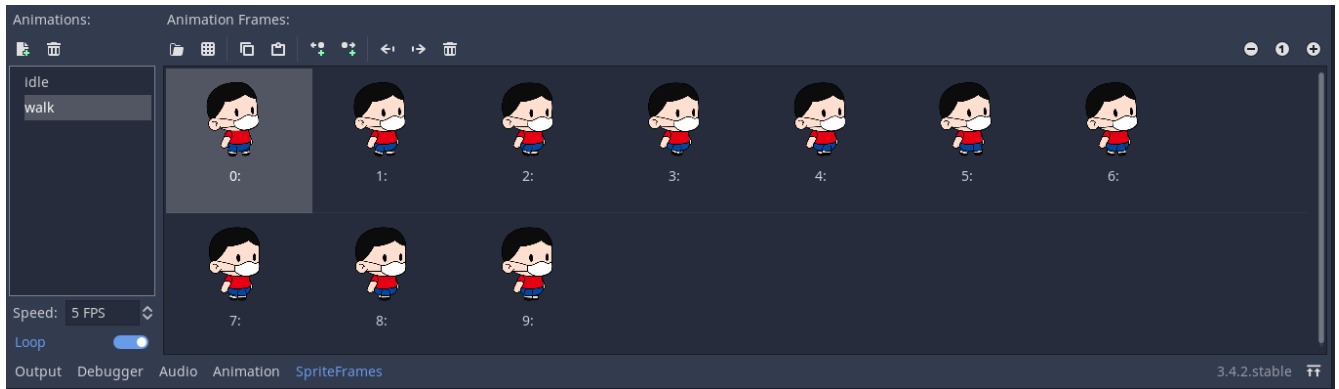


Gambar 15. Karakter dimasukkan dalam Animated Sprite

Karakter yang sudah dibuat dimasukkan ke dalam *node AnimatedSprite* seperti pada Gambar 4.1. *Node AnimatedSprite* merupakan *node sprite* sekaligus animasi. Untuk membuat animasi baru, pada bagian kanan layar terdapat menu *inspector*. Menu *inspector* tersebut mendeskripsikan atribut apa saja yang dimiliki oleh node. Klik panah ke bawah pada atribut *Frames*, kemudian pilih *New SpriteFrames*. Setelah itu gambar karakter setiap frame satu per satu dimasukkan melalui ikon folder pada bagian *Animation Frames > Add Texture from File*. Untuk mengganti nama animasi, pada bagian kiri Animation Frames, terdapat menu Animations lalu klik nama animasi yang ingin diubah. Untuk menambah animasi baru, pada menu Animations di sebelah kiri terdapat ikon kertas. Klik ikon tersebut, beri nama untuk animasi baru, lalu kembali ke *Animation Frames > Add Texture from File* dan hasil akan seperti pada Gambar 16 dan Gambar 17. Kemudian diberi *CollisionShape2D* sebanyak 3 buah yang fungsi untuk mengambil collectible item, mendeteksi musuh yang menabrak atau ditabrak karakter dengan menaruh *CollisionShape2D* pada *Area2D* yang diberi nama *EnemyDetector* dan agar karakter dapat menginjak virus dengan menaruh *CollisionShape2D* pada *Area2D* yang diberi nama *StompDetector*. Setelah itu, buatlah *script* untuk menggerakkan karakter, tambahkan fungsi dari *PlayerData.gd* lalu, lakukan *attach script*. Semua metode implementasi ini berlaku sama untuk karakter virus dan kerumunan.



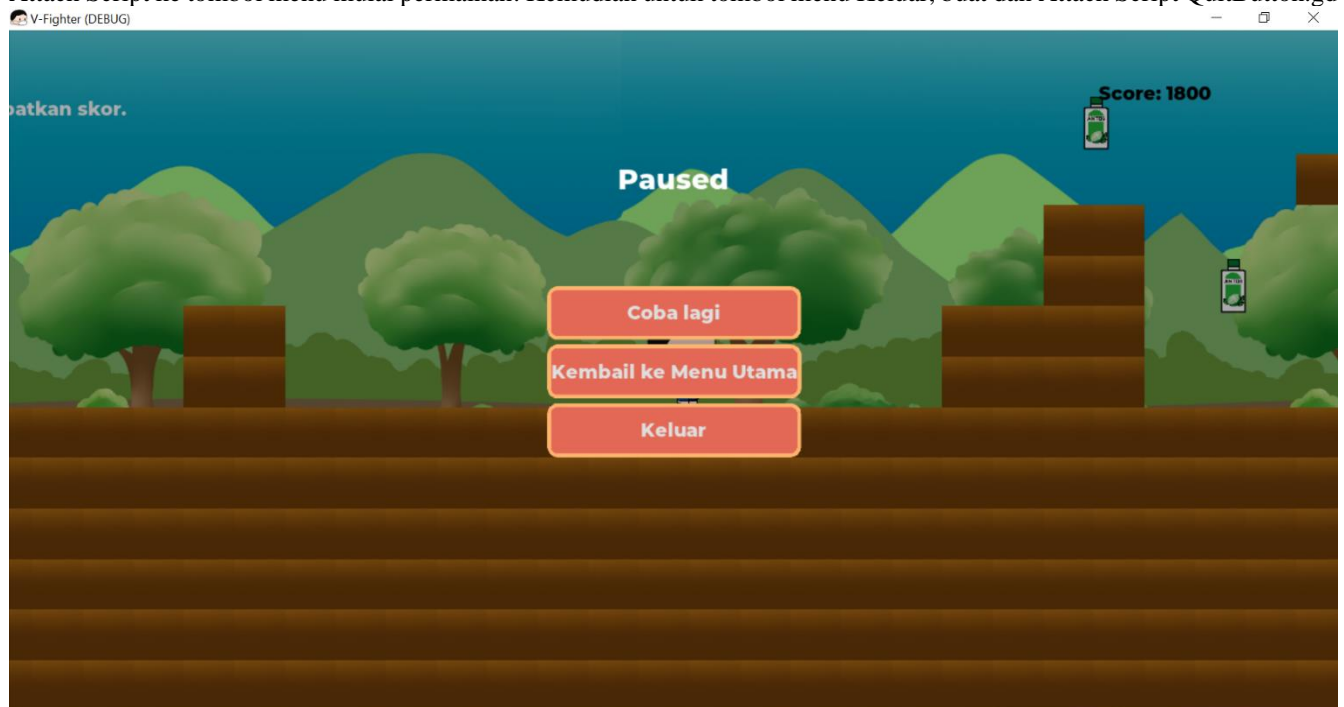
Gambar 16. Animasi diam



Gambar 17. Animasi bergerak

B. Implementasi Perpindahan Menu

Untuk dapat berpindah dari satu *scene* ke *scene* lainnya, maka dibuatlah skrip `SceneChangeButton.gd` yang kemudian di Attach Script ke tombol menu mulai permainan. Kemudian untuk tombol menu Keluar, buat dan Attach Script `QuitButton.gd`



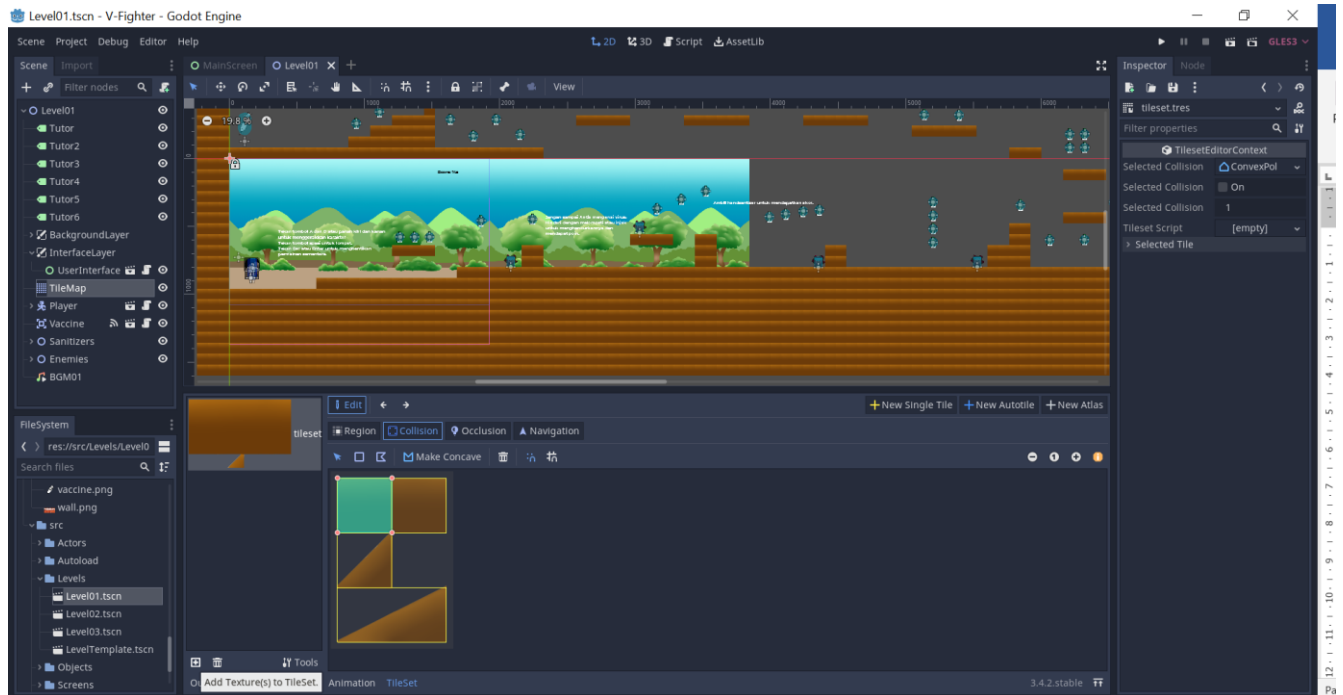
Gambar 18. Menu Pause

Menu pause pada Gambar 18 memiliki 3 tombol yaitu coba lagi, kembali ke menu utama dan keluar yang akan muncul pada saat tombol enter ditekan. Semua diatur dalam `UserInterface.tscn` dengan `node UserInterface` sebagai `node` induk yang di dalamnya terdapat `child node label Score` yang akan menerima input skor dari `hand sanitizer` dan `virus`, lalu `child node Pause Overlay` yang akan muncul pada saat fungsi `set_paused` bernilai `true` dan `AudioStreamPlayer` yang audionya akan diputar pada saat fungsi `_on_Player_died()` terjadi. Untuk membuat menu dan skor berfungsi, Attach Script `UserInterface.gd` pada `node induk UserInterface`.

C. Pembuatan Permainan

Setelah semua elemen yang diperlukan sudah terkumpul, susun elemen-elemen tersebut ke dalam sebuah *scene* baru. Kemudian buat `tilemap` pada masing-masing *scene* (karena level ada 3 buah maka total ada 3 `tilemap`). Pilih `Add Child Node > Node2D > TileMap`, lalu lihat inspector di sebelah kanan, cari `Tile Set` dan pilih `New TileSet`, klik `TileSet`, klik ikon kotak bertanda tambah untuk menambah tekstur, pilih gambar yang akan digunakan. Setelah memilih gambar, pada bagian kanan atas terdapat 3 menu untuk membuat `tile` diantaranya: `New Single Tile`, `New Auto Tile`, dan `New Atlas Tile`. Pilih `New Single`

Tile lalu seleksi gambar pada blok bagian bawah, kemudian klik *Collision* yang terletak di sebelah atas lalu ke menu *inspector* di bagian *Selected Collision*, klik *New ConvexPolygonShape2D*, hasilnya akan seperti Gambar 19 lalu susunlah ke dalam *scene*.



Gambar 19. Tilemap

Setelah selesai menyusun tilemap, tarik dan lepas *Player.tscn*, *Enemy.tscn*, *Crowd.tscn*, *Sanitizer.tscn* dan *Vaccine.tscn* ke dalam *scene* level dan atur penempatannya. Untuk tampilan skor dan menu pause, tambahkan *child node* baru bernama *InterfaceLayer*, lalu tarik dan lepas *UserInterface.tscn*. Untuk membuat level 02 dan 03 juga melakukan metode yang sama.

D. Cara Bermain

Langkah-langkah untuk menjalankan permainan adalah sebagai berikut:

- Pastikan aplikasi permainan ada pada komputer.
- Buka aplikasi permainan.
- Setelah splash screen, akan muncul tampilan awal menu mulai permainan dan keluar.
- Pilih menu mulai permainan untuk mulai bermain.
- Pemain memasuki level awal permainan (level 01). Dalam level ini, pemain diberi petunjuk tentang cara bermain dan pemain harus menyelesaikan hingga level 03.
- Setelah berhasil menyelesaikan semua level, permainan berakhir dan masuk ke tampilan akhir yang menampilkan jumlah skor, jumlah mati, dan menu berupa main lagi, ke menu utama, dan keluar.

V. PENGUJIAN

A. User Acceptance Test

Pengujian permainan menggunakan *User Acceptance Test* dengan jenis *Alpha* dan *Beta Testing*. *Alpha Testing* dilakukan untuk mencari *bug* dan masalah lain sebelum permainan dirilis untuk anak-anak. Pengujian *Alpha Testing* sudah dilakukan terhadap 3 responden dan mendapatkan hasil sebagai berikut:

TABEL I
PENGUJIAN ASPEK PERMAINAN

Aspek Pengujian	Sangat Tidak Puas	Kurang Puas	Cukup	Puas	Sangat Puas
Perasaan setelah bermain			2	1	
Edukasi protokol kesehatan			1	1	1
Mekanisme gerak <i>user</i>			1	2	
Desain tatap muka		1	2		
Desain karakter			1	2	
Desain <i>collectible item</i>			1	2	
Desain latar arena			1	1	1
Efek suara		1	2		
Suara latar		1	1	1	

TABEL II
HASIL PENGUJIAN *BUG/ERROR*

Hasil Pengujian <i>Bug/Error</i>			Keterangan Hasil Uji
Ya	2	66,7%	Terdapat <i>bug/error</i>
Tidak	1	33,3%	Tidak terdapat <i>bug/error</i>

Jumlah persen dari hasil pengujian *bug/error* pada Tabel II, didapatkan dari rumus sebagai berikut:

$$\frac{\text{Jumlah yang menjawab "Ya"}}{\text{Total responden}} \times 100\% \text{ untuk jawaban ya,}$$

$$\frac{\text{Jumlah yang menjawab "Tidak"}}{\text{Total responden}} \times 100\% \text{ untuk jawaban tidak}$$

Pada hasil pengujian *bug/error* pada Tabel II, sebanyak 33,3 persen merespon tidak menemukan *bug/error* (1 dari 3 responden menjawab “Tidak”) dan sebanyak 66,7 persen (2 dari 3 responden menjawab “Ya”) menunjukkan bahwa dalam permainan masih terdapat *bug/error* dengan rincian sebagai berikut:

- Virus yang menghilang dan skor tetap bertambah walau tidak diinjak oleh pemain (virus hanya mengenai kaki karakter pemain).
- Karakter utama melayang di ujung platform.

Bug tersebut berasal dari ukuran *collision shape* yang terletak pada karakter pemain kurang sesuai sehingga solusi untuk *bug* tersebut adalah mengubah ukuran *collision shape* menyesuaikan dengan karakter pemain. Kemudian diuji kembali dan setelah masalah *bug* berhasil terpecahkan, permainan dirilis ulang.

Untuk pengujian *beta testing*, permainan yang diujikan berupa permainan yang telah diperbaharui dari hasil uji coba *alpha testing*. Uji coba permainan dilaksanakan di SD Santo Yusup I Cikutra, kota Bandung dengan jumlah responden sebanyak 40 siswa rentang usia 6 sampai 10 tahun dipilih oleh wali kelas dari kelas 1 sampai dengan 4. Untuk sistem pengujian, siswa diminta untuk memainkan permainan yang sudah dibuat. Setelah selesai bermain, siswa akan mengisi kuesioner yang sudah disediakan. Pengujian *beta testing* pada 40 siswa terdiri dari pengujian aspek permainan (Tabel III), pengetahuan protokol kesehatan (Tabel IV), pendapat mengenai menyenangkan atau tidak permainan (Tabel V), bermanfaat atau tidak permainan untuk edukasi protokol kesehatan (Tabel VI), edukasi protokol kesehatan yang dapat diperoleh dalam permainan (Tabel VII) dan *bug/error* yang ditemukan dalam permainan (Tabel VIII). Hasil pengujian adalah sebagai berikut:

TABEL III
PENGUJIAN ASPEK PERMAINAN (BETA TESTING)

Aspek yang diuji	Sangat puas	Puas	Cukup	Kurang puas	Sangat tidak puas
Gerakan karakter	18	14	6	1	0
Desain menu	16	18	4	0	0
Desain karakter	22	10	5	1	0
Desain <i>collectible item</i>	14	21	1	1	1
Desain latar belakang	18	15	6	0	0

Desain arena permainan	18	15	5	2	0
------------------------	----	----	---	---	---

TABEL IV
PENGUJIAN PENGETAHUAN PROTOKOL KESEHATAN

Mengetahui Protokol Kesehatan		Keterangan Hasil Uji
Ya	39	Sebanyak 39 responden mengetahui protokol kesehatan.
Tidak	1	1 responden tidak mengetahui protokol kesehatan.

TABEL V
PENGUJIAN PERASAAN BERMAIN V-FIGHTER

Permainan V-Fighter Menyenangkan		Keterangan Hasil Uji
Ya	34	Sebanyak 34 responden menyatakan permainan terasa menyenangkan.
Cukup	5	Sebanyak 5 responden menyatakan permainan cukup menyenangkan.
Tidak	1	1 responden merasa permainan tidak menyenangkan karena merasa kesulitan menyelesaikan permainan.

TABEL VI
PENGUJIAN MANFAAT V-FIGHTER UNTUK EDUKASI PROTOKOL KESEHATAN

Permainan V-Fighter Bermanfaat Untuk Edukasi Protokol Kesehatan		Keterangan Hasil Uji
Ya	36	Sebanyak 36 responden menyatakan bahwa V-Fighter bermanfaat untuk edukasi protokol kesehatan.
Cukup	4	Sebanyak 4 responden menyatakan permainan V-Fighter cukup bermanfaat untuk edukasi protokol kesehatan.
Tidak	0	0 responden menjawab Tidak

TABEL VII
PENGUJIAN EDUKASI PROTOKOL KESEHATAN YANG TERDAPAT PADA V-FIGHTER

Edukasi Protokol Kesehatan Dalam Permainan		Keterangan Hasil Uji
Menggunakan masker saat berkegiatan di luar rumah	32	Sebanyak 32 responden menyatakan adanya edukasi protokol kesehatan untuk menggunakan masker (seperti karakter utama) dalam permainan.
Mencuci tangan atau menggunakan hand sanitizer	19	Sebanyak 19 responden menyatakan adanya edukasi protokol kesehatan untuk menggunakan hand sanitizer.
Menjaga jarak atau menghindari berkerumun	15	15 responden menyatakan adanya edukasi protokol kesehatan menghindari berkerumun.

TABEL VIII
PENGUJIAN BUG/ERROR PERMAINAN V-FIGHTER

Hasil Pengujian Bug/Error		Keterangan Hasil Uji
Ya	3	Sebanyak 3 responden menemukan bug/error dalam permainan.
Tidak	37	37 responden tidak menemukan bug/error dalam permainan.

VI. KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan maka peneliti dapat mengambil kesimpulan bahwa perangkat lunak Godot dapat digunakan untuk menciptakan permainan *platformer 2D* yang dapat memberikan edukasi akan pentingnya protokol kesehatan bagi anak-anak. Aplikasi telah berjalan sesuai rancangan, namun masih terdapat banyak ruang untuk pengembangan lebih lanjut. Contohnya memberi persyaratan skor untuk vaksin sehingga dapat pindah ke level berikutnya, memperlambat gerakan kamera yang masih terlalu cepat, menambah level lebih banyak lagi, mempercantik tampilan level dan karakter, memperbanyak musuh, virus dapat melakukan gerakan lain selain berjalan dan permainan dapat dirilis ke playstore.

DAFTAR PUSTAKA

- [1] "Monitoring Kepatuhan Protokol Kesehatan | Covid19.go.id." <https://covid19.go.id/monitoring-kepatuhan-protokol-kesehatan> (accessed May 30, 2022).
- [2] "Apa itu Video Game? - Refreshing / Brain Game - Dictio Community." <https://www.dictio.id/t/apa-itu-video-game/160579> (accessed May 06, 2022).
- [3] T. Bailey, F. Thabtah, M. Wright, and D. A. Tran, "FoodKnight: A mobile educational game and analyses of obesity awareness in children," *Health Informatics Journal*, vol. 26, no. 3. SAGE Publications Ltd, pp. 1684–1699, Sep. 01, 2020. doi: 10.1177/1460458219888405.
- [4] I. Wayan *et al.*, "FLAT DESIGN SEBAGAI GAYA VISUAL ASSET GAME 'ATTACK OF THE KRONA,'" 2022, [Online]. Available: <https://bit.ly/jurnalbahasarupa>
- [5] "Apa Itu Genre Game Platformer? - Esportsnesia." <https://esportsnesia.com/penting/apa-itu-genre-game-platformer/> (accessed May 06, 2022).
- [6] "Introduction to Godot — Godot Engine (stable) documentation in English." https://docs.godotengine.org/en/stable/getting_started/introduction/introduction_to_godot.html (accessed May 06, 2022).
- [7] "Scenes and nodes — Godot Engine (3.1) documentation in English." https://docs.godotengine.org/en/3.1/getting_started/step_by_step/scenes_and_nodes.html (accessed Jun. 02, 2022).
- [8] "Physics introduction — Godot Engine (stable) documentation in English." https://docs.godotengine.org/en/stable/tutorials/physics/physics_introduction.html#kinematicbody2d (accessed Jun. 02, 2022).
- [9] "Sprite — Godot Engine (stable) documentation in English." https://docs.godotengine.org/en/stable/classes/class_sprite.html (accessed Jun. 02, 2022).
- [10] "Using TileMaps — Godot Engine (stable) documentation in English." https://docs.godotengine.org/en/stable/tutorials/2d/using_tilemaps.html# (accessed Jun. 03, 2022).
- [11] "Membuat beragam Style Karakter Chibi | IkuZo! Japanese Education Center." <https://goikuzo.com/membuat-beragam-style-karakter-chibi/> (accessed Oct. 24, 2022).