

UAS
GAME DESAIN AND DEVELOPMENT
GAME CHESS 3D



Dosen : Firdaus Nur Sugiarto, S. Tr.Kom.

Disusun oleh :

Nama : Ilham Bintang Pratama

NIM : 17090022

Kelas : 6C

PROGRAM STUDI D4 TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA TEGAL
2020

Daftar isi

Cover	i
Daftar isi	ii
Tentang game	1
Konsep dan Pengembangann	2
Pengembangan Desain	3
Dokumentasi game	4
Dokuemntasi Source code	6

Tentang game

Game merupakan salah satu media hiburan yang banyak diminati oleh masyarakat. Pengembangan genre Game yang begitu pesat, menyebabkan Game semakin menarik banyak masyarakat. Setiap jenis Game memiliki keunggulan dan kekurangan masing-masing. Game Chess 3D yang dibangun ini merupakan jenis Game yaitu Sports, dengan melibatkan berbagai macam pola kecerdasan buatan yang ditanamkan pada non-player character (NPC). Metode pengembangannya adalah dengan pendekatan berorientasi objek dimulai dengan mengembangkan modal-modal dan environment yang ada. Penggambaran skema objeknya menggunakan Unified Modelling Language (UML) sebagai pemodelan aplikasi. Berdasarkan hasil pengujian yang dilakukan dapat disimpulkan bahwa, aplikasi permainan Chess telah berhasil mengimplementasikan Artificial intelligence Pada NPC. dan dapat menjadi sarana hiburan.

Non-Playable Character adalah jenis agent otomatis yang ditunjukkan untuk penggunaan komputer animasi, games dan virtual reality. Keberadaan agen ini mewakili tokoh dalam cerita atau permainan terkadang memiliki kemampuan untuk improvisasi tindakan mereka. Ini adalah kebalikan dari seorang tokoh dalam sebuah film animasi, yang tindakannya ditulis di muka, dan untuk “avatar” dalam sebuah game atau virtual reality, tindakan yang diarahkan secara real time oleh player.

FSM adalah sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga hal berikut: State (Keadaan), Event (kejadian) dan action (aksi). Pada satu saat dalam periode waktu yang cukup signifikan, sistem akan berada pada salah satu state yang aktif. Sistem dapat beralih atau bertransisi menuju state lain jika mendapatkan masukan atau event tertentu, baik yang berasal dari perangkat luar atau komponen dalam sistemnya itu sendiri. Transisi keadaan ini umumnya juga disertai oleh aksi yang dilakukan oleh sistem ketika menanggapi inputan yang terjadi. Aksi yang dilakukan tersebut dapat berupa aksi yang sederhana atau melibatkan rangkaian proses yang relatif kompleks.

Unity adalah salah satu game engine yang cukup terkenal dan banyak digunakan. Selain karena gratis, game engine ini tergolong mudah digunakan dan memiliki banyak referensi yang bisa digunakan sebagai bahan pembelajaran. Tujuan dari pembuatan game Sports Chess 3D ini adalah bagaimana cara agar dapat mengimplementasikan Artificial Intelligence pada Non-Player Character. Manfaat penelitian yang dapat diperoleh dari penelitian ini ialah memberikan alternatif sebagai sarana media hiburan interaktif dan sebagai sarana penghilang stres. Dapat menjadi sebagai kontribusi dan motivasi pengembangan dan pembuatan game 3D yang khususnya bergenre Sports.

Konsep dan Pengembangan

- Konsep

Tahap konsep yaitu menentukan tujuan aplikasi (informasi, hiburan, pelatihan dll), identifikasi penggunaan (user), bentuk aplikasi (presentasi, interaktif, dll) dan spesifikasi umum (ukuran spesifikasi, dasar perancangan, target yang ingin dicapai, dll). Pada tahap ini ditentukan bagian apa saja yang dapat dijadikan pedoman dalam pembuatan game.

Beberapa deskripsi konsep dari game yang dibuat adalah sebagai berikut:

a. Judul

Perancangan Game Bergenre Sports untuk Platform PC Games.

b. Pengguna

Para pengguna diharuskan minimal berumur lima belas tahun dan tidak disarankan untuk anak dibawah umur karena mengandung beberapa adegan menyeramkan serta tidak disarankan untuk orang yang mempunyai riwayat penyakit jantung karena terdapat beberapa scene jumpscare. Pengguna game diasumsikan pernah menggunakan komputer dan mengerti Bahasa Indonesia.

c. Tujuan

Tujuan dari pembuatan game ini adalah memberikan metode melatih diri dalam mengasah otak.

d. Jenis game

Jenis game yang dibuat oleh penulis adalah Role Playing Game yang dimana game bergenre sports.

e. Teknik pembuatan game

Penulis membuat game menggunakan teknik hierarki tiga dimensi yang dimana pembuatan lokasinya menggunakan aplikasi Blender dan diimport ke kedalam Unity 5 3D serta penambahan Assets (Villain, Texture, Sound, Prefabs, 3D Model, dll) bersumber dari Assets Store.

f. Gambar

Gambar yang digunakan penulis dalam pembuatan game adalah file yang berformat .JPG dan .png.

g. Audio

Format audio yang digunakan penulis dalam pembuatan game adalah file berformat .mp3 dan .wav.

h. Software authoring

Penggunaan software Blender dalam pembuatan lokasi/area permainan dan Unity 5 3D sebagai game engine utama yang digunakan.

Pengembangan Design

a. Game Element

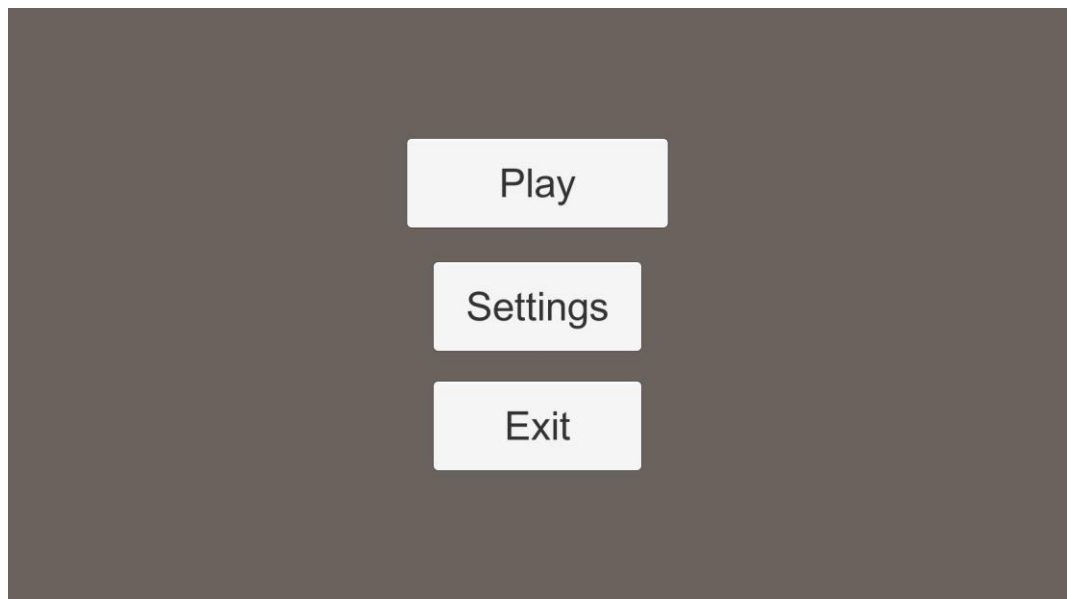
Game berjudul “Chess 3D” adalah sebuah game Role Playing Game yang bergenre Sports yang dirancang khusus untuk platform Desktop (Windows, Linux, dan Mac). Sistem interaksi antar pemain dan villain menggunakan time interval realtime yaitu dimana pemain dapat dengan secara langsung bergerak, menghindar tanpa harus menunggu bergantian (turn based).

b. Narasi Game

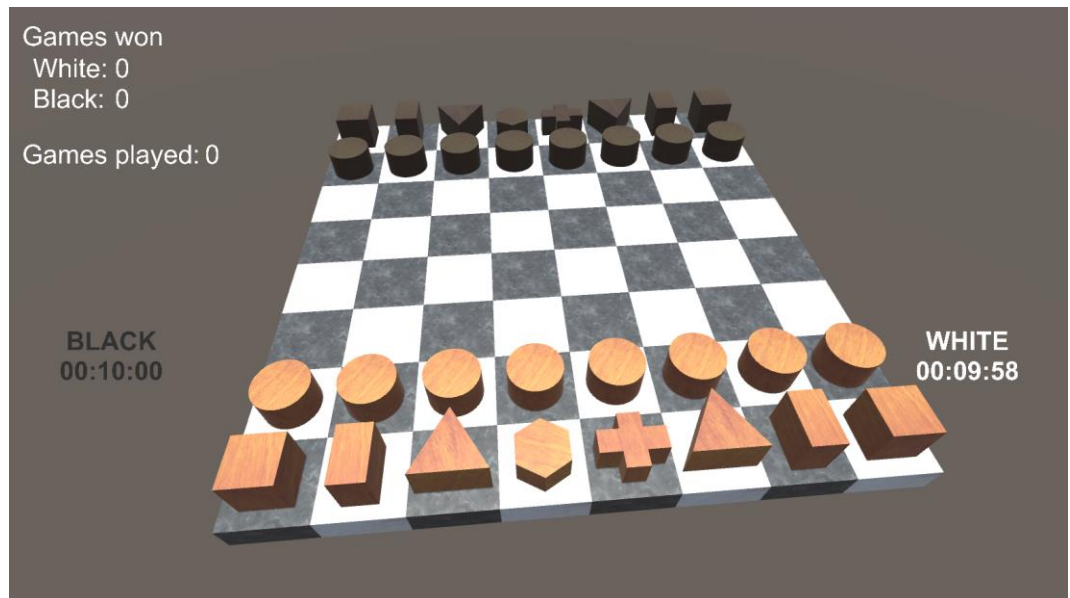
Chess 3D adalah permainan papan strategi dua orang yang dimainkan pada sebuah papan kotak-kotak yang terdiri dari 64 kotak, yang disusun dalam petak 8×8, yang terbagi sama rata (masing-masing 32 kotak) dalam kelompok warna putih dan hitam.

c. Gameplay

Game Chess 3D hanya dapat dimainkan secara multiplayer, dimana dua pemain dihadapkan dalam satu papan catur dan saling bertanding satu lawan satu (1vs1)



Gambar 1. Main Menu



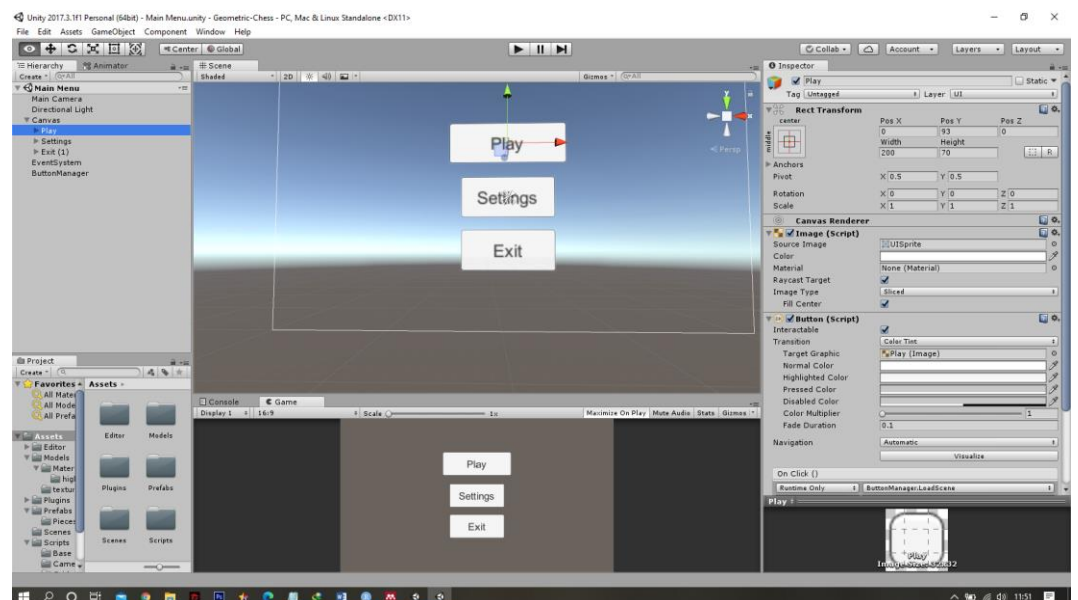
Gambar 2. Game Mulai

- **Pembuatan**

Dokumentasi game

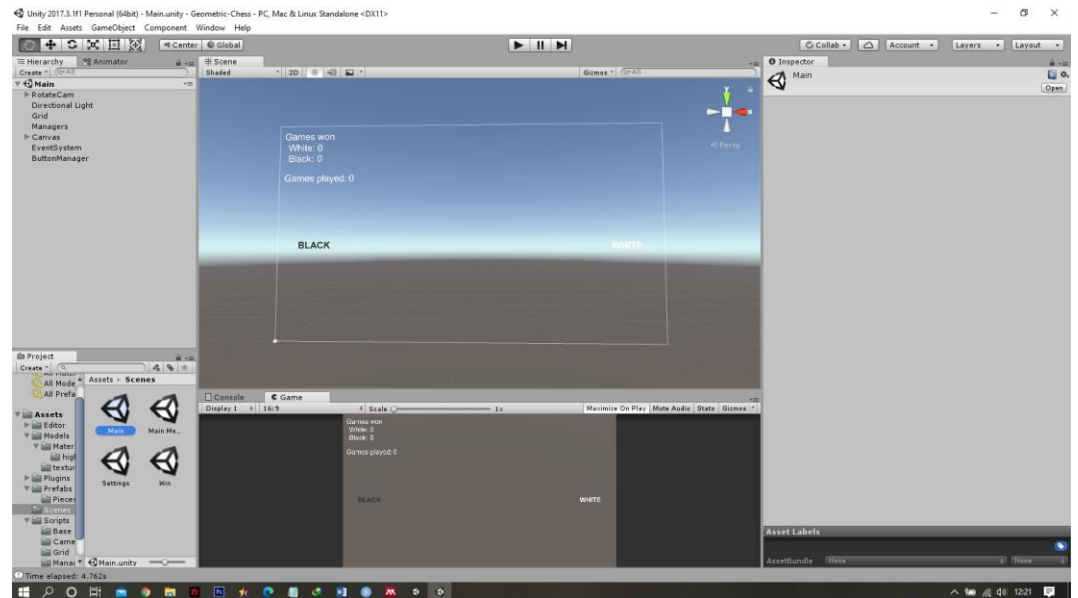
1. Main Menu

Play button, Setting button, dan Exit button



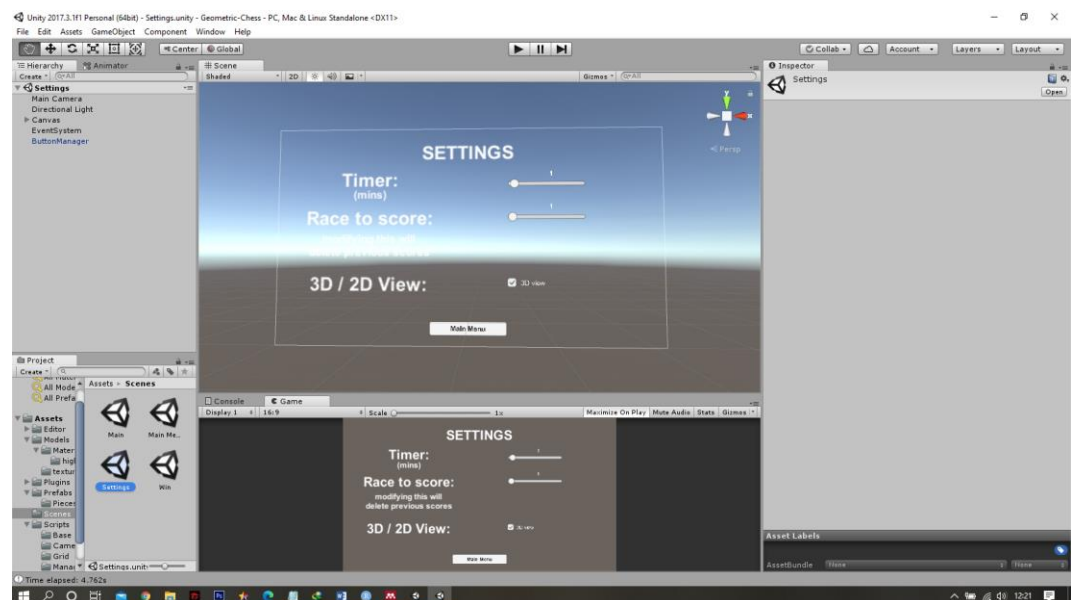
Gambar 3. Doc Main Menu

2. Gameplay (Play button)



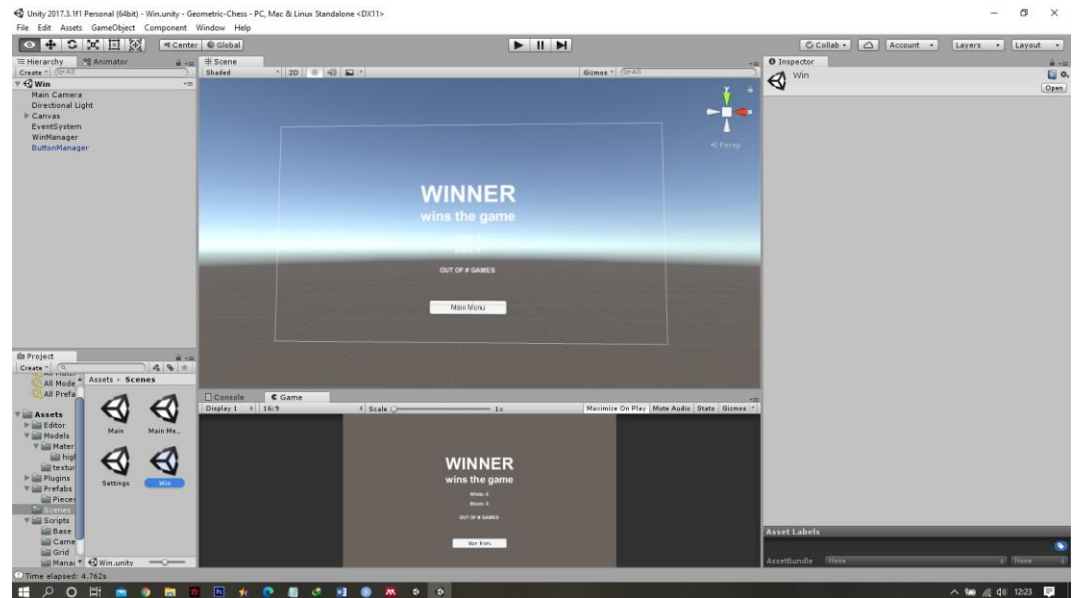
Gambar 4 Doc Gameplay

3. Setting View (Setting button)



Gambar 5. Doc Setting View

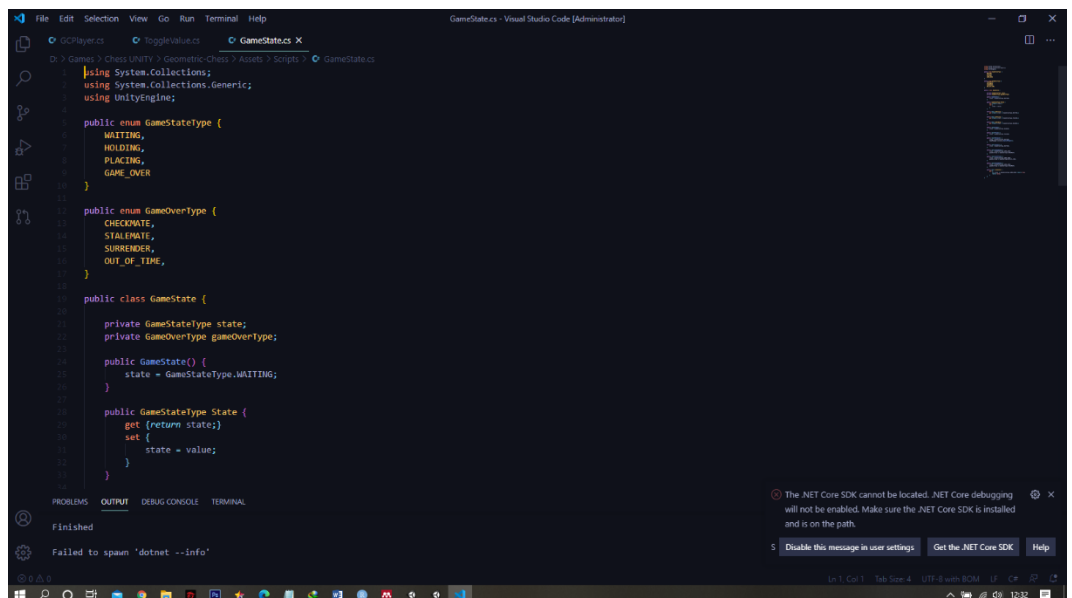
4. Winner View



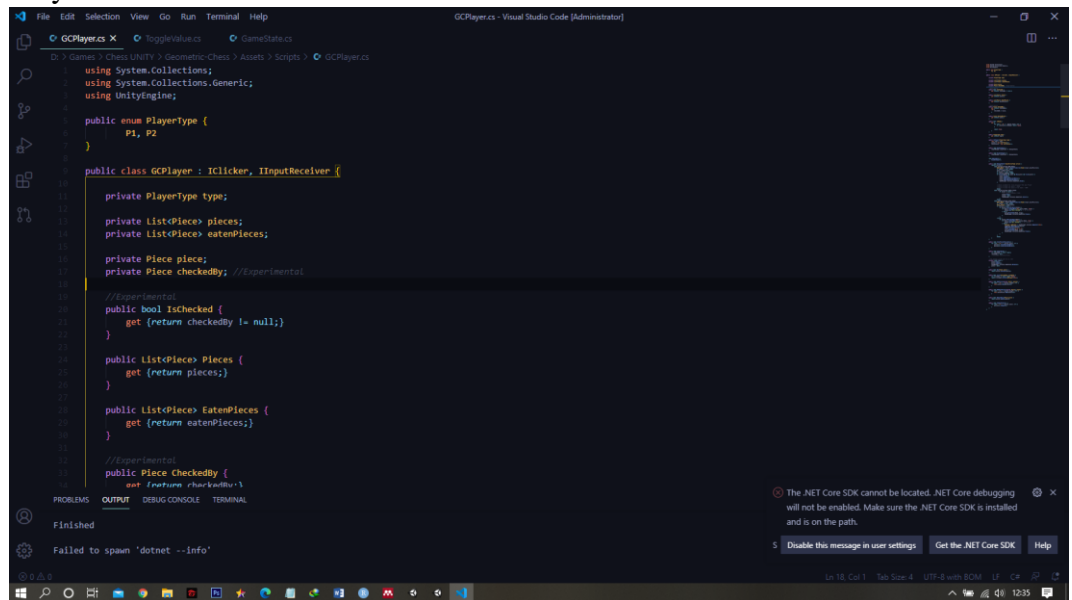
Gambar 6. WinnerView

Dokumentasi Source code

1. Game State

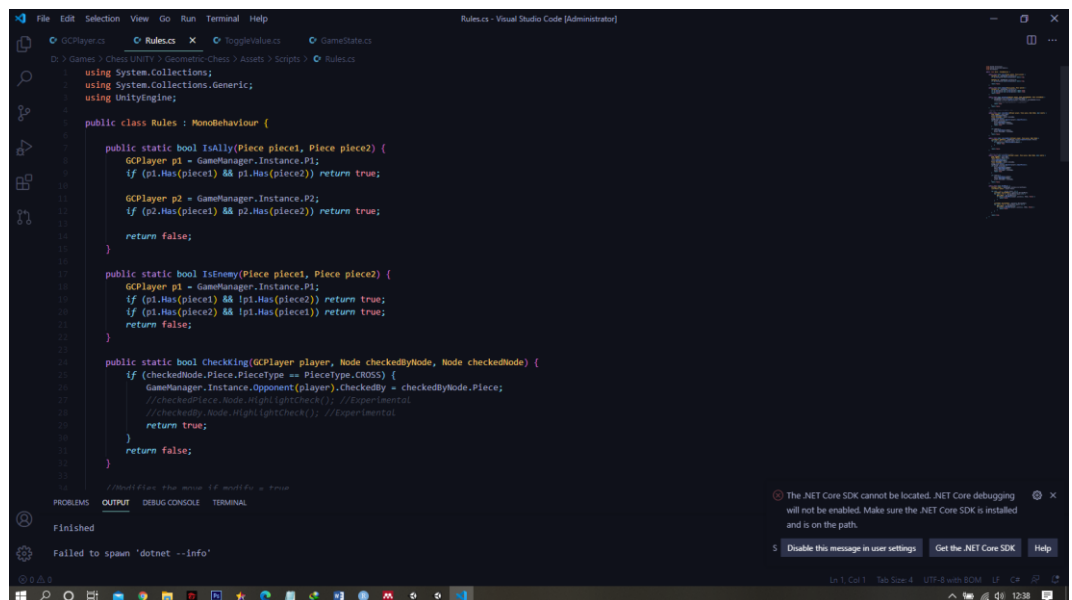


2. Player



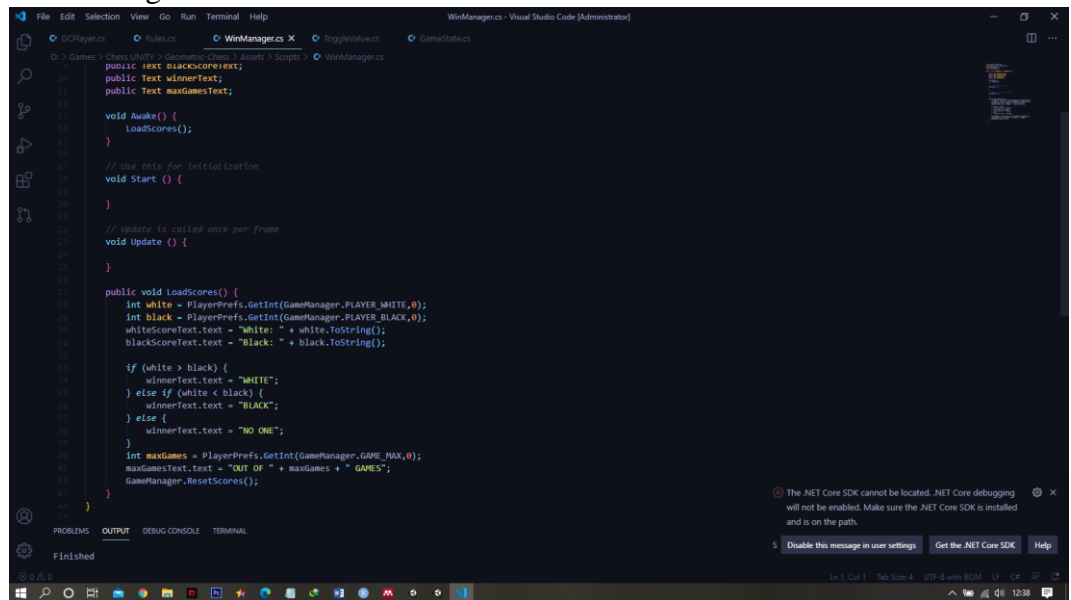
```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public enum PlayerType {
6     P1, P2
7 }
8
9 public class GCPlayer : IClicker, IInputReceiver {
10
11     private PlayerType type;
12     private List<Piece> pieces;
13     private List<Piece> eatenPieces;
14
15     private Piece piece;
16     private Piece checkedBy; //Experimental
17
18     //Experimental
19     public bool IsChecked {
20         get { return checkedBy != null; }
21     }
22
23     public List<Piece> Pieces {
24         get { return pieces; }
25     }
26
27     public List<Piece> EatenPieces {
28         get { return eatenPieces; }
29     }
30
31     //Experimental
32     public Piece CheckedBy {
33         get { return checkedBy; }
34     }
35 }
```

3. Rules



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Rules : MonoBehaviour {
6
7     public static bool IsAlly(Piece piece1, Piece piece2) {
8         GCPlayer p1 = GameManager.Instance.P1;
9         if (p1.Has(piece1) && p1.Has(piece2)) return true;
10
11         GCPlayer p2 = GameManager.Instance.P2;
12         if (p2.Has(piece1) && p2.Has(piece2)) return true;
13         return false;
14     }
15
16     public static bool IsEnemy(Piece piece1, Piece piece2) {
17         GCPlayer p1 = GameManager.Instance.P1;
18         if (p1.Has(piece1) && !p1.Has(piece2)) return true;
19         if (p1.Has(piece2) && !p1.Has(piece1)) return true;
20         return false;
21     }
22
23     public static bool CheckKing(GCPlayer player, Node checkedByNode, Node checkedNode) {
24         if (checkedNode.Piece.PieceType == PieceType.KING) {
25             GameManager.Instance.Opponent(player).CheckedBy = checkedByNode.Piece;
26             //checkedByNode.HighLightCheck(); //Experimental
27             //checkedByNode.HighLightCheck(); //Experimental
28             return true;
29         }
30         return false;
31     }
32 }
```

4. Win Manager



The screenshot shows the Visual Studio Code editor with the `WinManager.cs` file open. The file is located at `D:\Games > Chess UNITY > Geometric/Chess > Assets > Scripts > WinManager.cs`. The script contains the following C# code:

```
1  public Text blackScoreText;
2  public Text winnerText;
3  public Text maxGamesText;
4
5  void Awake() {
6      LoadScores();
7  }
8
9  // Use this for initialization
10 void Start () {
11 }
12
13 // Update is called once per frame
14 void Update () {
15 }
16
17 public void LoadScores() {
18     int white = PlayerPrefs.GetInt(GameManager.PLAYER_WHITE, 0);
19     int black = PlayerPrefs.GetInt(GameManager.PLAYER_BLACK, 0);
20     whiteScoreText.text = "White: " + white.ToString();
21     blackScoreText.text = "Black: " + black.ToString();
22
23     if (white > black) {
24         winnerText.text = "WHITE";
25     } else if (white < black) {
26         winnerText.text = "BLACK";
27     } else {
28         winnerText.text = "NO ONE";
29     }
30
31     int maxGames = PlayerPrefs.GetInt(GameManager.GAME_MAX, 0);
32     maxGamesText.text = "OUT OF " + maxGames + " GAMES";
33     GameManager.ResetScores();
34 }
35 }
```

The bottom of the window shows the `PROBLEMS` panel with a message: "The .NET Core SDK cannot be located. .NET Core debugging will not be enabled. Make sure the .NET Core SDK is installed and is on the path." Below this message are buttons for "Disable this message in user settings", "Get the .NET Core SDK", and "Help". The status bar at the bottom indicates "Ln 1, Col 1", "Tab Size: 4", "UTF-8 with BOM", and "C#".