

Proyek Akhir OOP

Kelompok 14

Andhika Fadhlán Wijanarko 2306267164

Adrian Dika Darmawan 2306242376

Jonathan Matius 2306161896



AiPlayer

- **Inisialisasi:**
 - Konstruktor AiPlayer menginisialisasi nama AI.
 - Menyimpan kartu di tangannya dalam sebuah daftar bernama handList.
- **Logika Giliran** (turn() method):
 - Menganalisis kartu paling atas di tumpukan buangan (currDisc) untuk menentukan kartu yang valid dimainkan.
 - Menghitung jumlah kartu di tangan berdasarkan warna, angka, dan jenis khusus (wild atau wild draw four).
 - Memutuskan kartu terbaik untuk dimainkan dengan prioritas:
 - **Match warna > Match angka > Mainkan kartu khusus.**
 - Jika tidak ada kartu valid, AI akan mengambil kartu dari tumpukan.
- **Pemilihan Warna untuk Kartu Wild:**
 - Memilih warna yang paling banyak dimiliki dalam tangan.
 - Jika hanya memiliki kartu wild berwarna hitam, AI memilih warna secara acak yang berbeda dari warna kartu di tumpukan buangan.
- **Memainkan Kartu:**
 - Jika kartu dimainkan, status permainan diperbarui, dan pesan dikirimkan melalui objek Control.
 - Menangani efek kartu spesial seperti **skip**, **reverse**, **draw 2**, **wild**, dan **wild draw 4**.
- **Implementasi Antarmuka Pemain:**
 - Mengimplementasikan metode yang diperlukan oleh antarmuka PlayerInterface, seperti getName(), getCardsLeft(), dan addCards().

Card

- **Konstruktor**
- `csharp`
- Salin kode
- ```
public Card(int numb, string color, GameObject obj)
```
- - Menginisialisasi kartu dengan nomor, warna, dan objek prefab.
- **`loadCard(int x, int y, Transform parent)`**
  - Fungsi untuk menempatkan kartu pada posisi tertentu di layar.
  - **Paramete r:**
    - `x, y` - Koordinat posisi kartu.
    - `parent` - Transformasi induk (parent) dari kartu.
  - **Mengembalikan:** Objek kartu (`GameObject`) yang sudah diinstansiasi.
- **`loadCard(Transform parent)`**
  - Sama seperti metode sebelumnya, tetapi digunakan jika kartu tidak memerlukan posisi spesifik.
  - Kartu akan diinstansiasi sebagai anak (child) dari transformasi induk yang diberikan.
- **`returnColor(string what)`**
  - Fungsi internal untuk mengubah nama warna (string) menjadi nilai warna (Color32) yang dapat digunakan Unity.
  - Contoh:
    - "Green" menjadi hijau (0x55AA55).
    - "Blue" menjadi biru (0x5555FD).
- **`getNum()`**
  - Akses ke properti nomor kartu (number).
- **`getColor()`**
  - Akses ke properti warna kartu (color).
- **`Equals(Card other)`**
  - Override metode bawaan Equals untuk membandingkan dua kartu.
  - Kartu dianggap sama jika:
    - Nomornya sama, atau
    - Warnanya sama.
- **`changeColor(string newColor)`**
  - Mengubah warna kartu (biasanya untuk kartu Wild atau Wild Draw 4).
  - Contoh: Mengubah kartu Wild menjadi warna "Red" setelah pemain memilih warna merah.

# Control

- **Atribut Utama:**
- **List<PlayerInterface>:** Menyimpan pemain (baik manusia maupun AI).
- **static List<Card>:**
  - **deck:** Menyimpan semua kartu yang tersedia untuk dimainkan.
  - **discard:** Menyimpan kartu yang sudah dimainkan (di "discard pile").
- **GameObject:**
  - Objek UI seperti tangan pemain (**playerHand**), warna pilihan, tombol pause, dll.
- **Prefabs:**
  - Digunakan untuk membuat kartu dengan tipe yang berbeda (**skip**, **reverse**, **wild**, dll.).
- **Properti Status:**
  - Seperti giliran saat ini (**where**), arah permainan (**reverse**), dan jumlah AI.
  - **Metode Utama:**
- **Start():**
  - Melakukan inisialisasi permainan: membuat pemain, menyusun deck, dan memulai game.
- **Update():**
  - Menangani giliran pemain (baik manusia maupun AI).
- **shuffle():**
  - Mengacak urutan kartu dalam deck.
- **specialCardPlay():**
  - Menangani efek kartu spesial seperti **skip**, **reverse**, **draw**, atau **wild**.
- **draw():**
  - Memberikan kartu kepada pemain dari deck.
- **resetDeck():**
  - Mengisi ulang deck dari discard pile saat deck habis.
- **startWild()** dan **addWildListeners():**
  - Menangani pemilihan warna setelah kartu **wild** dimainkan.
- **updateCardsLeft():**
  - Memperbarui jumlah kartu yang tersisa untuk setiap pemain, serta mengecek jika ada pemenang.
- **playAgain():**
  - Mengatur ulang permainan untuk memulai kembali.



# HumanPlayer

- **skip:** Menyimpan status apakah pemain harus dilewati pada giliran berikutnya (misalnya, jika kartu "Skip" dimainkan).
- **drew:** Menyimpan status apakah pemain telah menarik kartu pada giliran tersebut.
- **playedWild:** Menyimpan status apakah pemain telah memainkan kartu Wild.
- **name:** Nama pemain.
- **handList:** Daftar kartu yang dimiliki pemain.

# Menu

- **Tujuan:** Memberikan fungsi menu utama, seperti menampilkan kartu, mengatur konfigurasi permainan, dan memulai permainan.
- **Komponen UI:**
  - Kartu dengan warna dan angka acak.
  - Efek kartu yang bergerak (scrolling).
  - Tombol toggle untuk memilih jumlah pemain AI.
  - Layar pengaturan (setup canvas) untuk konfigurasi permainan.
- **Komponen Utama**
- **Atribut**
  - **Elemen UI:**
    - `versi`: Menampilkan versi game.
    - `cards`: Array objek UI untuk menampilkan kartu acak.
    - `setupCanvas`: Mengontrol visibilitas layar pengaturan.
    - `toggle`: Array tombol toggle untuk memilih jumlah pemain AI.
  - **Prefab Kartu:**
    - `regCardPrefab`, `skipCardPrefab`, `revrsCardPrefab`, `drawCardPrefab`, `wildCardPrefab`: Prefab untuk jenis kartu yang berbeda.
  - **Kartu Bergerak:**
    - `scroll1`: Array objek untuk efek kartu bergerak.
    - `scroll2`: Daftar objek yang dikelola secara dinamis untuk efek scrolling kartu.
- **Metode**
  - **Start():**
    - Inisialisasi menu, termasuk memberikan warna dan angka acak pada kartu.
    - Membuat kartu awal untuk efek scrolling.
  - **Update():**
    - Mengelola efek scrolling kartu dan mereset kartu yang sudah keluar dari layar.
  - **Helper (Fungsi Pendukung):**
    - `returnRandColor(int rand)`: Mengembalikan warna kartu secara acak dalam bentuk string.
    - `returnColor(string what)`: Mengonversi nama warna menjadi objek `Color` untuk styling UI.
  - **Opsi Menu:**
    - `exit()`: Menutup aplikasi.
    - `setUp(bool openClose)`: Mengontrol visibilitas layar pengaturan.
    - `play()`: Menentukan jumlah pemain AI berdasarkan toggle yang dipilih dan memulai permainan dengan berpindah ke layar utama.



# PlayerInterface

- **turn()**
- Metode tanpa nilai kembali.
- Mengatur logika untuk giliran pemain. Setiap pemain harus memiliki implementasi spesifik untuk menjalankan langkah mereka.
- **skipStatus (Properti)**
- **Tipe:** bool (Boolean).
- Properti dengan getter dan setter.
- Menunjukkan apakah pemain harus dilewati pada giliran mereka (karena efek kartu tertentu, seperti kartu skip).
- **addCards(Card other)**
- Metode untuk menambahkan kartu ke tangan pemain.
- Parameter other adalah objek dari kelas Card, yang merepresentasikan kartu yang akan ditambahkan.
- **getName()**
- Mengembalikan nama pemain dalam bentuk string.
- Digunakan untuk mengidentifikasi pemain dalam game.
- **Equals(PlayerInterface other)**
- Membandingkan dua objek pemain untuk menentukan apakah mereka sama.
- Parameter other adalah objek pemain yang dibandingkan.
- **getCardsLeft()**
- Mengembalikan jumlah kartu yang tersisa di tangan pemain.
- Tipe pengembalian: int.