


```
{  
}
```

Card 클래스 - ConvertToString 메서드

```
public class Card {  
    // 멤버 변수  
    public string ConvertToString()  
    {  
        string result = "";  
  
        // 숫자  
        if (number >= 2 && number <= 10) // 2~10  
        {  
            result = $"{number}";  
        }  
        else if (number == 1) // A  
        {  
            result = "A";  
        }  
        else if (number == 11) // J  
        {  
            result = "J";  
        }  
        else if (number == 12) // Q  
        {  
            result = "Q";  
        }  
        else // K  
        {  
            result = "K";  
        }  
  
        // 문자  
        switch (suit) // "suit"  
        {  
            case CardSuit.Heart:  
                result += "♥";  
            case CardSuit.Spade:  
                result += "♠";  
            case CardSuit.Club:  
                result += "♣";  
            case CardSuit.Diamond:  
                result += "♦";  
        }  
    }  
}
```


Hand [] - CountSameNumber []

```
// [ ]
// number: [ ]1[ ]13[ ]
private int CountSameNumber(int number)
{
    // [ ]
    if (number < 1 || number > 13)
    {
        Debug.Log("[ ]!");
        return 0; // [ ]
    }

    int count = 0;
    foreach (Card c in cards)
    {
        if (c.Number == number)
            count++;
    }

    return count;
}
```

????

[] - CountSameNumber []

```
// [ ]
private bool IsOnePair()
{
}
```

Hand [] - IsOnePair []

```
// [ ]
private bool IsOnePair()
{
    // [ ]
}
```



```

// 同様に実装
if (CountSameCard(i) == 2 && i != match)
    return true;
}

return false;
}

```

????????????

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

// 同様に実装
private bool IsThreeOfAKind()
{
}

```

Hand 3 - IsThreeOfAKind 3

```

// 同様に実装
private bool IsThreeOfAKind()
{
    // 同様に実装
    for (int i = 1; i <= 13; i++)
    {
        if (CountSameCard(i) == 3)
            return 3;
    }

    return false;
}

```

?????

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

3

- XXXXXXXXXXXXXXXXXXXX
- XXXXXXXXXXXXXXXXXXXX
 - XXXXXXXX +0 → 1

Hand [] - CheckResult []

```
// [ ]
public void CheckResult()
{
    // [ ]
    string result = "";
    foreach (Card c in cards)
    {
        result += c.ConvertToString();
    }

    result += " - ";

    // [ ]→[ ]
    if (IsStraightFlush())
    {
        result += "[ ]";
    }
    else if (IsFourOfAKind())
    {
        result += "[ ]";
    }
    else if (IsFullHouse())
    {
        result += "[ ]";
    }
    else if (IsFlush())
    {
        result += "[ ]";
    }
    else if (IsStraight())
    {
        result += "[ ]";
    }
    else if (IsThreeOfAKind())
    {
        result += "[ ]";
    }
}
```

```

    }
    else if (IsTwoPair())
    {
        result += "TwoPair";
    }
    else if (IsOnePair())
    {
        result += "OnePair";
    }
    else
    {
        result += "NoPair";
    }

    // Output
    Debug.Log(result);
}

```

??????????

XXXXXXXXXXXXXXXXXXXX

XXXXXX _____

```

// TestPoker.cs

using UnityEngine;

public class TestPoker : MonoBehaviour
{
    void Start()
    {
        // Initialize Hand
        Hand hand = new Hand();

        // Set Cards
        hand.SetCard(0, CardSuit.Heart, 8);
        hand.SetCard(1, CardSuit.Club, 8);
        hand.SetCard(2, CardSuit.Diamond, 8);
        hand.SetCard(3, CardSuit.Heart, 11);
    }
}

```

```

hand.SetCard(4, CardSuit.Spade, 11);
hand.CheckResult();

// 同花
hand.SetCard(0, CardSuit.Heart, 5);
hand.SetCard(1, CardSuit.Club, 7);
hand.SetCard(2, CardSuit.Diamond, 8);
hand.SetCard(3, CardSuit.Heart, 6);
hand.SetCard(4, CardSuit.Spade, 9);
hand.CheckResult();

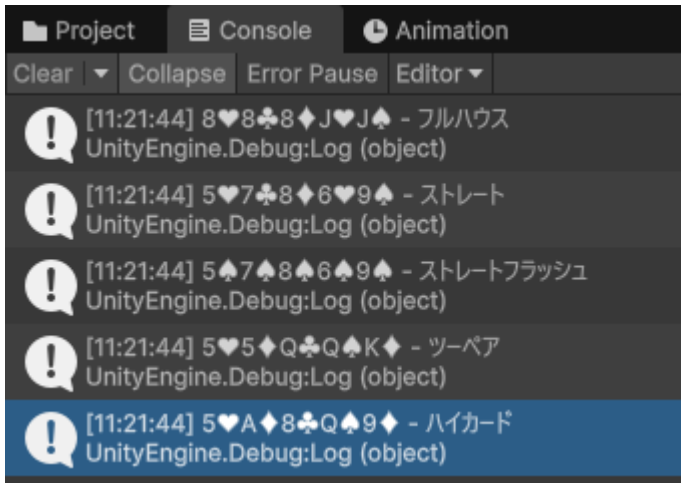
// 同花顺
hand.SetCard(0, CardSuit.Spade, 5);
hand.SetCard(1, CardSuit.Spade, 7);
hand.SetCard(2, CardSuit.Spade, 8);
hand.SetCard(3, CardSuit.Spade, 6);
hand.SetCard(4, CardSuit.Spade, 9);
hand.CheckResult();

// 对子
hand.SetCard(0, CardSuit.Heart, 5);
hand.SetCard(1, CardSuit.Diamond, 5);
hand.SetCard(2, CardSuit.Club, 12);
hand.SetCard(3, CardSuit.Spade, 12);
hand.SetCard(4, CardSuit.Diamond, 13);
hand.CheckResult();

// 对子
hand.SetCard(0, CardSuit.Heart, 5);
hand.SetCard(1, CardSuit.Diamond, 1);
hand.SetCard(2, CardSuit.Club, 8);
hand.SetCard(3, CardSuit.Spade, 12);
hand.SetCard(4, CardSuit.Diamond, 9);
hand.CheckResult();
}
}

```

?????



Revision #6

Created 2026-04-07 04:17:47 UTC by Menendez Francisco

Updated 2026-04-08 02:24:11 UTC by Menendez Francisco