



Chess League

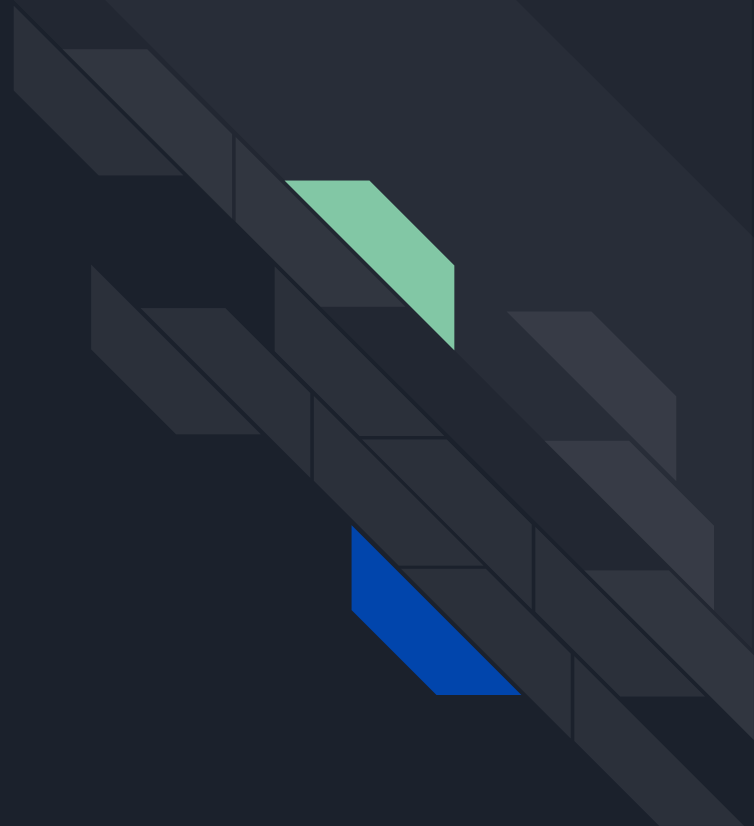
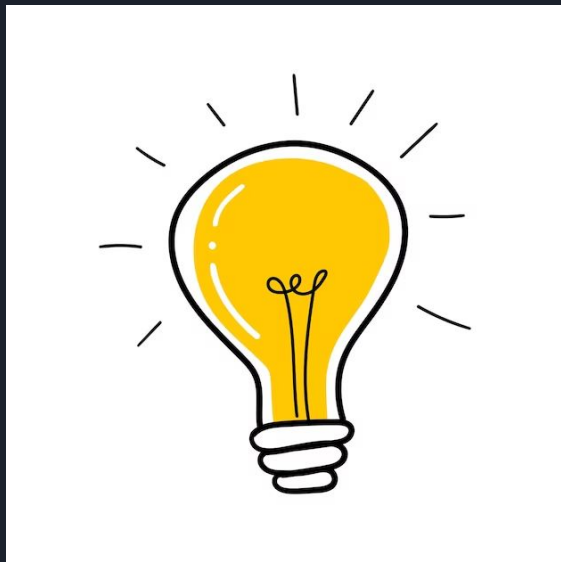
Eduardo Morell Muniz
Sergi Diaz Leiva

Introducció

- Perquè fem aquest projecte
- Motivacions personals
- Decidir per Unity

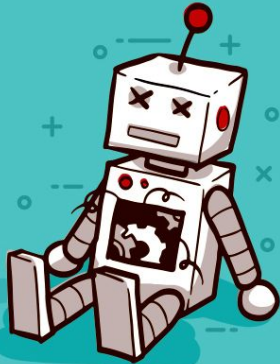


Plantejament i idees



Error Inicial

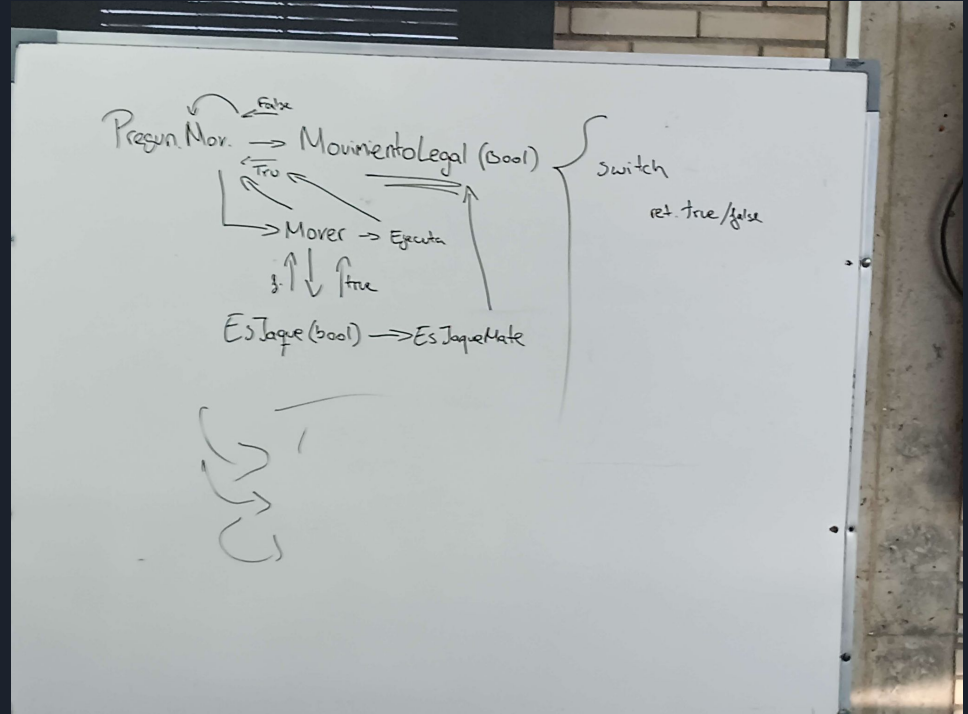
Oops!



- Començar a programar en interfície gràfica des del primer dia i online
- Cap creació d'esquema mental del joc
- Primera esquema en paper
- Conclusió de fer-ho consola
- Pensar Moviment Cada peça

Aclaració i direcció correcte

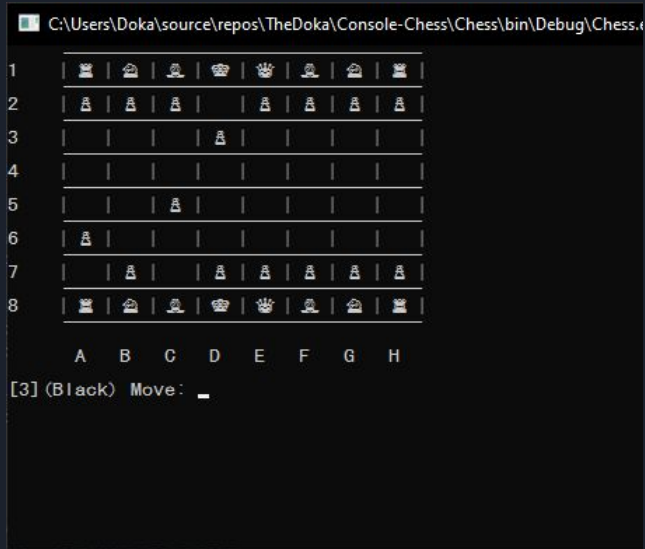
- Estructura de classes molt modular
- Necessitats de crear y saber com funcionen la lògica dels escacs
- Pensament Aplicació Cartes
- Probes Diverses maneres de jugar online Netcode, Riptide
- Error a futur, jaque amb mini bugs a consola



Primera Implementació amb Unity



Escacs el videojoc



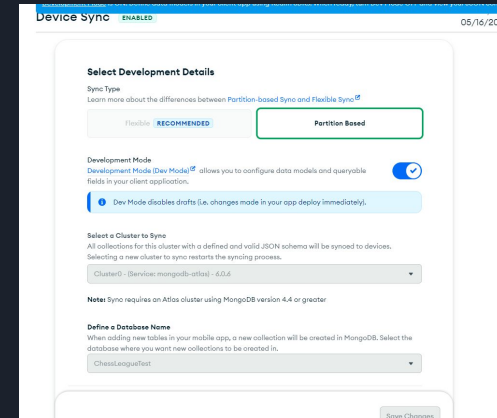
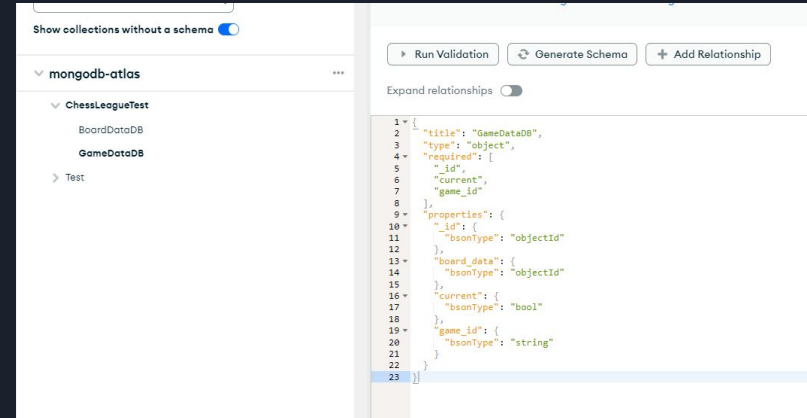
- Pas de c# consola a Unity
- Detecció entrada mouse i com interactua amb la taula(matriu escacs)
- Us intern i crides entre les diverses classes
- Problemes amb els jaques
- Connexió entre menus

Online i Base de Dades



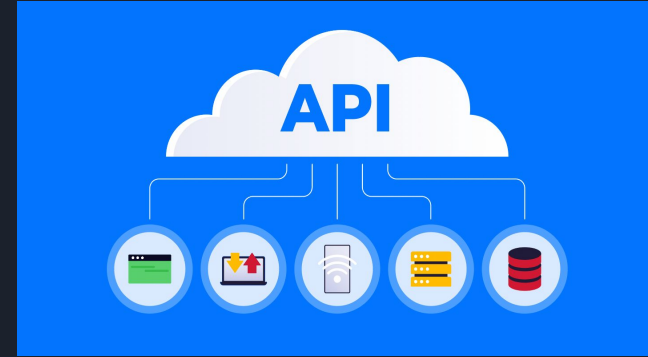
Base de Dades

- Decidir per MongoDB Realms
- Implementació MongoDB Realms a Unity
- Gestió Classes
- MongoDB Realms
- Funcionament esquemes(mapejat)
- Sync Partida
- Mètodes Autenticació
- Funcions aplicables extres



Implementació

- Testing Inicial
- Dificultats al testing, raig de llum
- Problemes Esquemes, particions, permisos, etc
- Diverses idees crear partida online i connexió:
Id random, Id per usuari, etc
- Solució final
- Funcionament RealmController com a API
- Connexió



```
    GameDataB gameData = GetOrCreateData();
    return (string)gameData.GameId;
}

public string GetWhitePlayerId()
{
    GameDataB gameData = GetOrCreateData();
    return (string)gameData.Board.WhitePlayerId;
}

public string GetBlackPlayerId()
{
    GameDataB gameData = GetOrCreateData();
    return (string)gameData.Board.BlackPlayerId;
}

public void UploadBoard(string value)
{
    GameDataB gameData = GetOrCreateData();
    _realm.Write() => {
        {
            gameData.Board.StringBoard = value;
        }
    };
}

public void UploadBlackPlayerId(string value)
{
    GameDataB gameData = GetOrCreateData();
    _realm.Write() => {
        {
            gameData.Board.BlackPlayerId = value;
        }
    };
}

public void DeleteBlackCard(int cardId)
{
    GameDataB gameData = GetOrCreateData();

    string[] cards = gameData.Board.BlackCards.Split(',');
    string newCards = "";
    bool found = false;

    for (int i = 0; i < cards.Length; i++)
    {
        if (int.Parse(cards[i]) == cardId && !found)
        {
            found = true;
        }
        else
        {
            if (i == cards.Length - 1)
            {
                newCards += cards[i];
            }
            else
            {
                newCards += cards[i] + ",";
            }
        }
    }
}
```

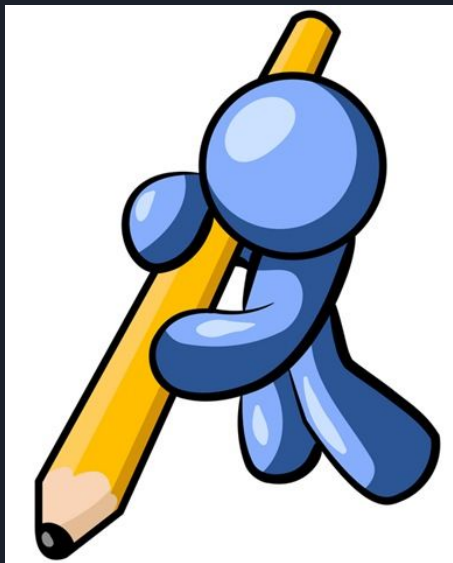
Cartes



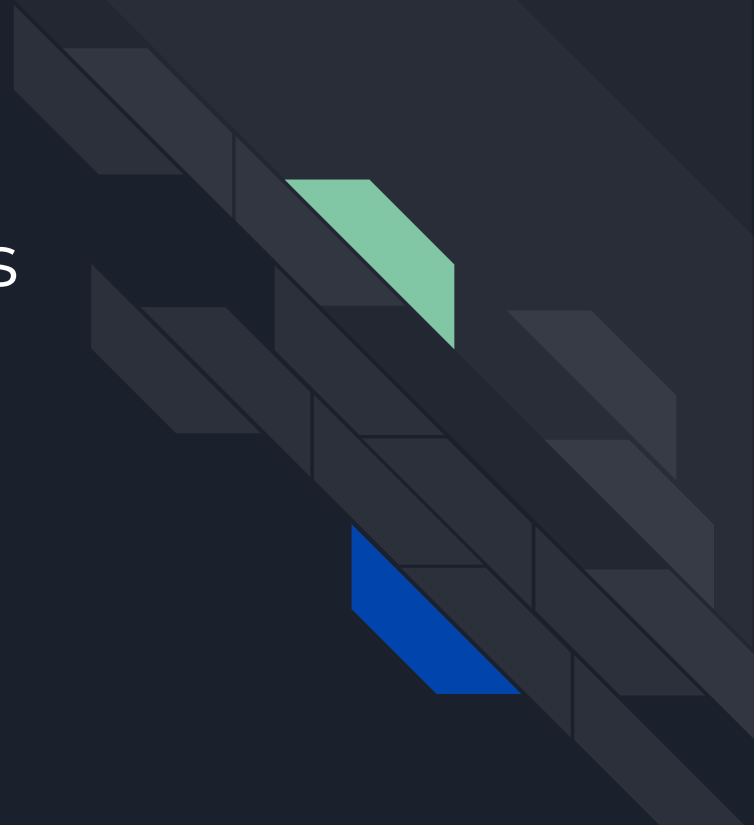
Implementació i visuals

- Disseny de les cartes
- Funcionalitats
- Implementació i persistència
- Animacions
- Efectes de so





Conclusions





- Joc Completat
- Temps Just
- Online a dins d'un joc aconseguit
- Funcionalitats no implementades com Google Ads o una classe jugador, menú, etc.
- Petits detalls d'últim moment diferència (so, efectes, configuració)
- Agradaria Continuar aquest joc