

FINAL PROJECT



08/05/24

Intermediate Programming with C# 2024

For Dor Ben Dor

David Daniel Shapiro

Jonathan Blau

Or Frenkel

The Engine

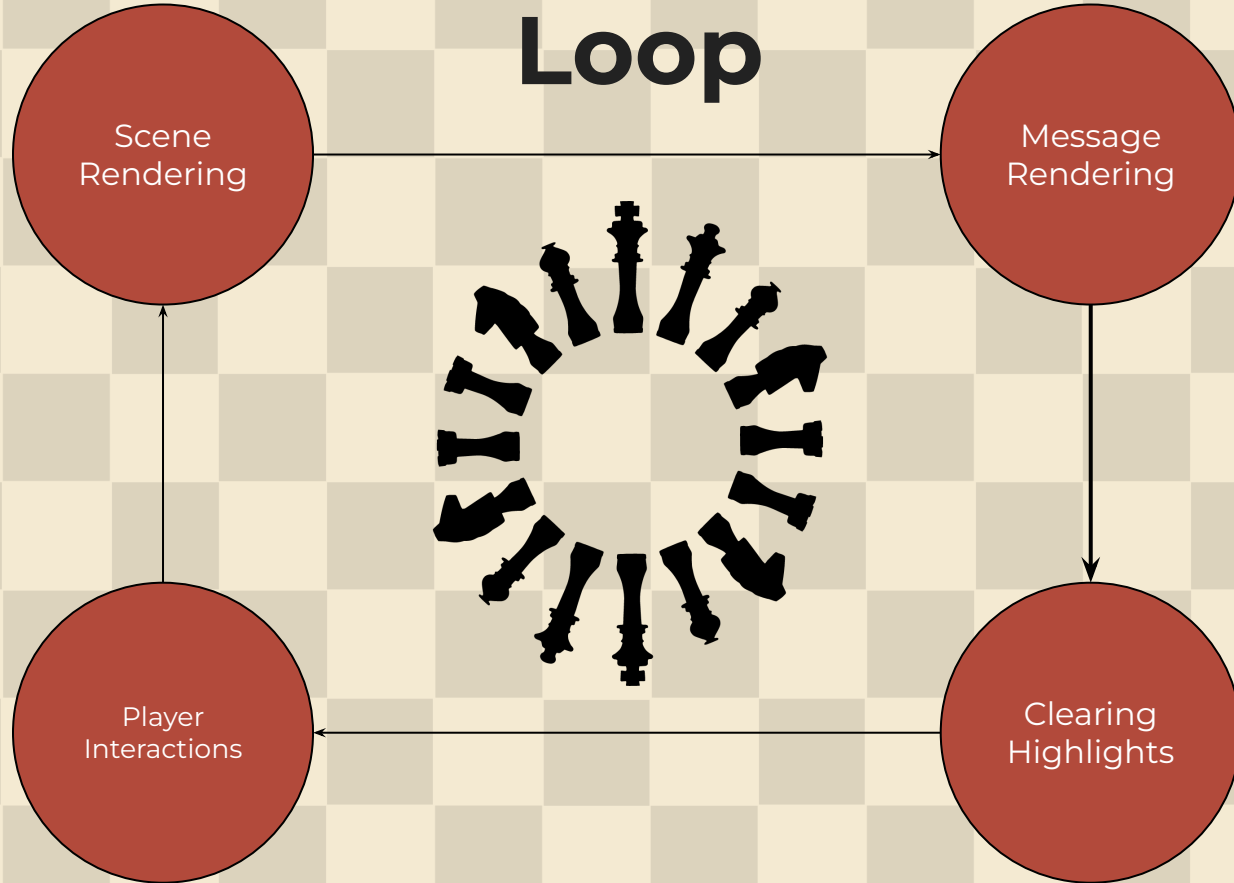
We decided to build an engine for:

CHESS

	A	B	C	D	E	F	G	H	
8	R	N	B	Q	K	B	N	R	8
7	P	P	P	P	P	P	P	P	7
6									6
5									5
4									4
3									3
2	P	P	P	P	P	P	P	P	2
1	R	N	B	Q	K	B	N	R	1
	A	B	C	D	E	F	G	H	

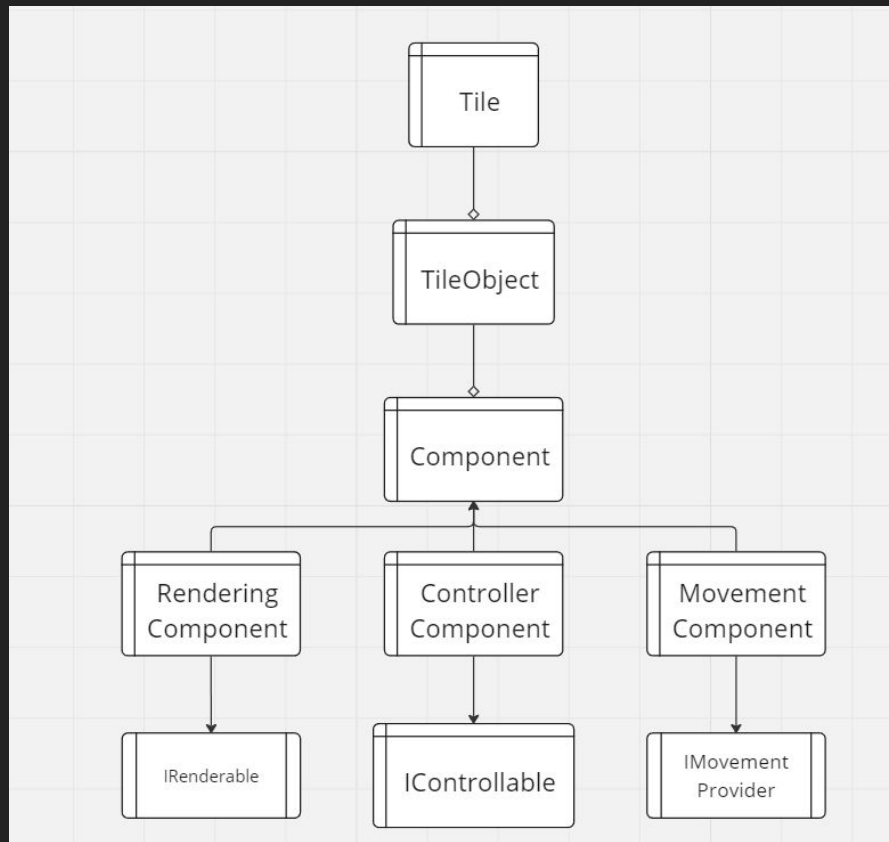
Blue Player turn

Engine Main Loop



Components

Simple to use, easy to extend



Asset Manager

Save your assets to json and load them anytime

```
// Load the assets
var RookPrefab = AssetManager.LoadAsset<Rook>("Rook");
var PawnPrefab = AssetManager.LoadAsset<Pawn>("Pawn");
var QueenPrefab = AssetManager.LoadAsset<Queen>("Queen");
var BishopPrefab = AssetManager.LoadAsset<Bishop>("Bishop");
var KnightPrefab = AssetManager.LoadAsset<Knight>("Knight");
var KingPrefab = AssetManager.LoadAsset<King>("King");

//Example of Adding assets to scene by reference to prefab component
Instantiate(RookPrefab, new(0, 0), Controllers[1]);
Instantiate(RookPrefab, new(7, 0), Controllers[1]);
Instantiate(QueenPrefab, new(3, 0), Controllers[1]);
Instantiate(BishopPrefab, new(2, 0), Controllers[1]);
Instantiate(BishopPrefab, new(5, 0), Controllers[1]);
Instantiate(KnightPrefab, new(1, 0), Controllers[1]);
Instantiate(KnightPrefab, new(6, 0), Controllers[1]);
Instantiate(KingPrefab, new(4, 0), Controllers[1]);
```

Commands

Command System is easy extendable
with whatever you want

```
CommandSystem.Instance.AddCommand(new SelectCommand("Select"));
CommandSystem.Instance.AddCommand(new DeselectCommand("Deselect"));
CommandSystem.Instance.AddCommand(new AttackCommand("Move"));
CommandSystem.Instance.AddCommand(new ShowCommand("Show"));
CommandSystem.Instance.AddCommand(new SelectAndMoveCommand("StockFish"));
CommandSystem.Instance.AddCommand(new FenCommand("FEN"));
```

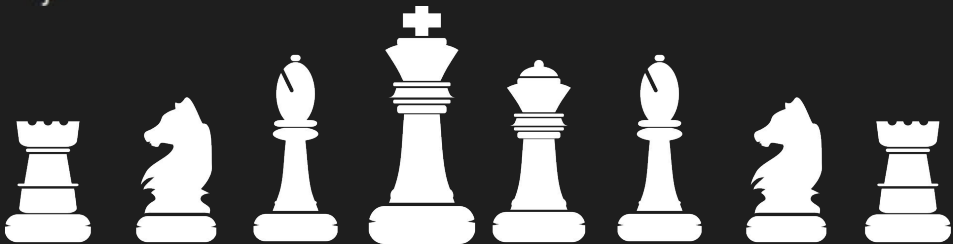
2 references

```
public void Listen(Func<string> action)
{
    if (currentAction != null) return;

    currentAction = action;
    action += (currentAction = null);
    var value = action();

    int activated = 0;
    string[] parameters = value.Split(' ');
    if (parameters.Length == 0) return;

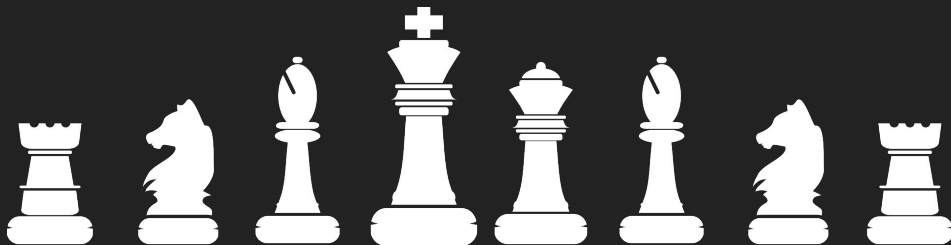
    foreach (var command in Commands)
    {
        if (parameters[0].ToLower() == command.Prompt.ToLower())
        {
            activated++;
            command.Activate(parameters);
        }
    }
    if (activated == 0)
    {
        HelpCommand.Activate(parameters);
    }
}
```



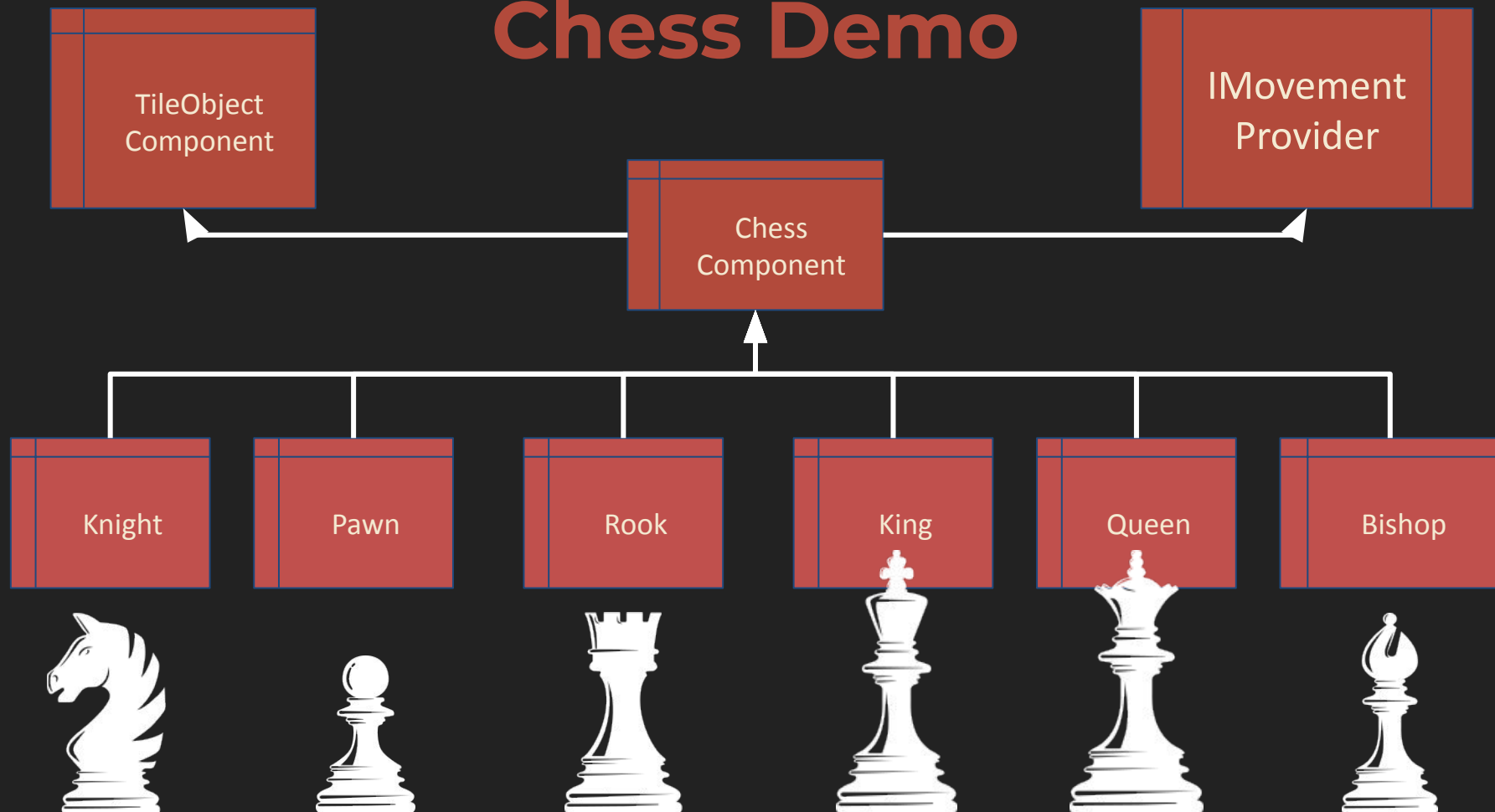
Renderer

The Renderer renders tiles to the console

- Borders
 - Numbers & Letters on the sides of the board
- Tiles
 - Background Colors
- Tile Objects
 - Letters
 - Foreground Colors
- Messages



Chess Demo



FilterForSelfCheck

- Goes over an array of all possible moves.
- Simulates them - Buuuuut doesn't save them
- Checks for checkmate.

```
private IEnumerable<Position2D> FilterForSelfCheck(IEnumerable<Position2D> attemptedMoves, Scene gameState, Position2D startPos, ChessActor actor)
{
    List<Position2D> notCheckMoves = new();
    var temp = gameState[startPos].TileObject;
    foreach (var move in attemptedMoves)
    {
        gameState[startPos].TileObject = null;
        var tempPiece = gameState[move].TileObject;
        gameState[move].TileObject = temp;
        temp.Position = move;
        if (actor.IsInCheck(gameState))
        {
            gameState[startPos].TileObject = temp;
            gameState[move].TileObject = tempPiece;
        }
        else
        {
            gameState[startPos].TileObject = temp;
            gameState[move].TileObject = tempPiece;
            notCheckMoves.Add(move);
        }
        temp.Position = startPos;
    }
    return notCheckMoves;
}
```

Show Command

[Masochist🏆]

	A	B	C	D	E	F	G	H	
8	R	N	B	Q	K	B		R	8
7	P	P	P		P	P			7
6						N	P		6
5				P				P	5
4				P					4
3			N		P		P		3
2	P	P	P			P		P	2
1	R		B	Q	K	B	N	R	1
	A	B	C	D	E	F	G	H	

Show in general

	A	B	C	D	E	F	G	H	
8	R	N	B	Q	K	B		R	8
7	P	P	P		P	P			7
6						N	P		6
5				P				P	5
4				P					4
3			N		P		P		3
2	P	P	P			P		P	2
1	R		B	Q	K	B	N	R	1
	A	B	C	D	E	F	G	H	

Show when selected

Chess Demo - AI

```
public static string ToFENFromat(this Scene scene)
```

	A	B	C	D	E	F	G	H	
8	R	N	B	Q	K	B	N	R	8
7	P	P	P	P	P	P	P	P	7
6									6
5									5
4									4
3									3
2	P	P	P	P	P	P	P	P	2
1	R	N	B	Q	K	B	N	R	1
	A	B	C	D	E	F	G	H	

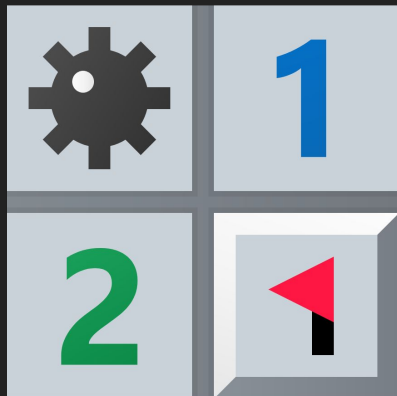
Blue Player turn

-

```
output += "- - 0 1";
```



MineSweeper Demo



	A	B	C	D	E	F	G	H	I	
9			1	1		1	0	1	F	9
8	F	F	1			2	1	1		8
7	2	2	1	1	2	F	1	1		7
6	0	0	0	0	2	2	2	1	F	6
5	1	1	0	0	2	F	2	1	1	5
4	F	1	0	0	2	F	2	0	0	4
3	1	1	0	0	1	1	1	0	0	3
2	0	0	0	0	0	0	0	0	0	2
1	0	0	0	0	0	0	0	0	0	1
	A	B	C	D	E	F	G	H	I	

D9 selected

flag e8
You Won!

Thank You!

