

Composition

Organiser koden slik at en metode har mange andre klasser som feltvariabel

Composition betyr at vi kun har en klasse Game

Game har feltvariabler som gir ulik funksjonalitet

Metodene i Game kaller da på metoder i field-variabelene

```
public class Game {  
    private GameBoard board;  
    private PlayerList players;  
    // etc ...  
    public Game(GameBoard board, PlayerLists players){  
        this.board = board;  
        this.players = new PlayerList();  
    }  
}
```

Også da har vi metoder i Game som kaller på andre ting for logikk.

```
private void nextPlayer(){  
    this.players.nextPlayer();  
}  
private void addPlayer(Player player){  
    this.players.add(player);  
}  
private boolean isWinner(Player player){  
    return winningRule.gameOver(board); // eksempel: enda en feltvariabel  
    winningrule  
}
```

Huskeregul: Is-a / has-a?

Tic-Tac-Toe *is a* game (therefore extends from Game)

Tic-Tac-Toe *has a* winning condition (therefore has a field variable winRule)

But..! All games *has a* winning condition (therefore has a field variable winRule)

Game Differences

Tetris is "Event-based"

Tic-Tac-Toe is a "Game loop" (multiple players)