

Boggle

Write Short units of code

Niet zo problematisch. Ik heb geprobeerd zoveel mogelijk functionaliteit op te delen.

Write Simple units of code

Er komen hier niet al te veel if statments in voor wat het makkelijker maakt. Er zijn ook niet meer dan twee loops nodig in één methode.

Write code once

Keep unit interfaces small

Om de interfaces wat beter te maken wrap ik de coördinaten van de letters in een object. Daarnaast wordt de code er een stuk makkelijker van

Separate concerns in module

Is niet in orde volgens bettercodehub. De twee controller classes spreken de board class meerdere malen aan. Vooral BoggleSearch is afhankelijk van Board. Verder is er te veel gekoppeld aan Coordinate class, die wordt gebruikt als een makkelijke manier om met de punten te werken. Ik kan de calls naar deze klass niet verminderen.

Couple Architecture Components Loosely

Wil dus totaal niet. Vijf van de acht klassen staan hier genoemd voor verbetering.

Keep Architecture Components Balanced

Ik ben blij met hoeveel beter het er uitziet vergeleken met de sorteren opdracht. De balans tussen model en controller is beter dan hiervoor.

Keep Your Codebase Small

Dit kan ik niet verkeerd doen.

Automate Tests

Simpelweg niet de moeite voor gedaan om het uit te zoeken en te implementeren.

Write Clean Code

Geen probleem meer aan het einde nadat ik alle klassen bij langs ben gegaan om onnodige comments weg te halen.

Algorithme

Voor het zoeken naar de woorden gebruik ik backtracking. Ik zoek door search() aan te roepen voor elke mogelijke punt als begin.

```
for (int i = 0; i < board.getSize(); i++) {  
    for (int j = 0; j < board.getSize(); j++) {  
        search(dictionary, new Coordinate(i, j), "");  
    }  
}
```

In de search methode wordt de gegeven locatie van het bord gemarkeerd als gebruikt zodat deze niet een tweede keer in hetzelfde woord zal worden gepakt. De aanliggende velden worden opgevraagd en deze worden een voor een bekenen voor het maken van woorden, verder wordt hier gekeken of de huidige letters al een woord vormen. Als het een unique woord is dan wordt deze toegevoegd aan de lijst. Als er een woord is met de huidige letters in deze volgorde als begin dan wordt de methode recursief aangeroepen om verder te zoeken.

Write Short units of code

Write Simple units of code

Write code once

Keep unit interfaces small

Separate concerns in module

Couple Architecture Components Loosely

Keep Architecture Components Balanced

Keep Your Codebase Small

Automate Tests

Write Clean Code