

In deze derde week pas je weer een aantal design patterns toe in ons al bestaande xonix-spel. Het gaat daarbij deze keer om de patterns: Proxy en Strategy. Deze zijn allebei besproken in het afgelopen college.

Je gaat verder werken aan het programma waarin je al een klassenstructuur voor de gameobjecten hebt gemaakt en de patterns Singleton, Observer en Command al hebt geïmplementeerd. Door de patterns Proxy en Strategy wordt het programma veiliger en wordt de functionaliteit vergroot.

Opdracht 1.

In *GameWorld* bestaat de klasse *State*. Een object van deze klasse wordt meegegeven aan *ScoreView* van *GameView*. *ScoreView* heeft volledige toegang tot *State*. Om die toegang te beperken en het programma dus veiliger te maken, verander je de klasse *State* als volgt. Maak eerst een interface *State* die alle veilige methodes bevat. Dat zijn die methodes die de toestandsvariabelen van *State* niet kunnen veranderen. Maak twee klassen die deze interface implementeren: *ProxyState* en *RealState*. In *ProxyState* worden alleen de veilige methodes geïmplementeerd. *RealState* is gelijk aan de oude klasse *State*. Die bevat naast alle veilige methodes ook de methodes die de toestandsvariabelen veranderen. Geef *GameWorld* toegang tot alle methodes via een *RealState*. Maak bij deze *RealState* ook een *ProxyState* die je meegeeft aan *ScoreView*, zodat die niets kan veranderen.

Extra optie: Het Proxy pattern kun je natuurlijk ook toepassen op *GameWorld* in zijn geheel in combinatie met proxy's voor de gameobjecten. Hiermee maak je het programma nog veiliger.

Opdracht 2.

Pas in de game ook het *Strategy* pattern toe. Doe dat voor de *MonsterBalls*. Deze *MonsterBalls* verplaatsen zich nu alleen nog in vaste richting en snelheid over het speelveld. Geef *MonsterBalls* ook andere bewegingsstrategieën mee. Je kunt daarbij denken aan een strategie waarbij een *MonsterBall* de *Car* achtervolgt. Bedenk ook nog een andere strategie. Deze strategieën moeten dynamisch (dus in runtime) kunnen worden gewijzigd. Dit kan goed door toepassing van het *Strategy* pattern. Geef nieuwe *MonsterBalls* verschillende strategieën mee. Wijzig de strategieën van de *MonsterBalls* door het commando *Switch Bomb Strategies* in het menu of door de spatiebalk in te drukken.

Opdracht 3.

Ga weer verder met het verbeteren van het programma. Vervolmaak zo mogelijk de implementatie van de patterns van vorige week en besteed ook weer aandacht aan de goede werking van de userinterface.

Tot slot

Na uitwerking van deze opdrachten heb je weer meer geleerd over objectgeoriënteerd programmeren in Java, begrijp je het gebruik van het framework Swing beter en kun je de patterns Proxy en Strategy toepassen.

De deadline voor de opdrachten van deze tweede week is vrijdag 2 december 2016. Je kunt je oplossing laten goedkeuren door de practicumdocent en uploaden via BlackBoard.