Design Patterns SS 2015
Hausaufgabe 7

Die Aufgaben müssen von jedem Teilnehmer einzeln bearbeitet und abgegeben werden. Die Abgabe muss **bis spätestens Montag 01.06.2014 um 23:59 Uhr** über unser Hausaufgabenverwaltungssystem **https://se.cs.uni-kassel.de/hms/** erfolgen (nicht per Email). Die Abgabe ist nur als einzelne *.zip Datei möglich. Daher müssen alle für eine Abgabe relevanten Daten zu einem solchen Archiv kombiniert werden.

Hinweise zur Abgabe:

- Die Hausaufgabe ist in Form von **einem** exportierten Eclipse Projekten abzugeben. Mit Hilfe der Eclipse Export Funktion ist es möglich mehrere Projekte **in einer zip-Datei** zu exportieren. Ist ein Projekt nicht korrekt exportiert, kann es bei der Korrektur nicht berücksichtigt werden (es bietet sich also an, den Import des exportierten Projektes auszuprobieren).

  WICHTIG: Benennen Sie ihre Projekte für diese Abgabe nach folgendem Schema:

  DPSS15_HA<a>_<b>_<Matrikelnummer>,

  wobei <a> für die aktuelle Hausaufgabe und <b> für die Aufgabennummer steht.
  Beispiel für Aufgabe 1:
  DPSS15_HA3_1_12345678.
- Libraries müssen in einen lib oder im Projekt eingefügt werden oder als weitere Projekt mit dem Namen DPSS15_HA<a>_libs_<Matrikelnummer>. Es darf keine Abhängigkeit zu anderen Projekten bestehen oder zu externen Libraries.

Allgemeines:
Orientiert euch für die Lösung der Aufgaben an den zugehörigen Übungen und Vorlesungen:
http://seblog.cs.uni-kassel.de/category/ss15/designpatternss15/

Zusammensetzung der Note:
Es wird n Hausaufgaben geben, von denen **n-2 Sinnvoll** bearbeitet werden müssen.
Sinnvoll bedeutet mindestens 50% der Punkte.
Am Ende ergibt sich die Note aus dem Durchschnitt n-2 besten Abgaben.
Zusätzlich wird es zwei Zusatzaufgaben geben mit denen fehlenden Abgaben ausgeglichen werden können.

Die Zusatzaufgaben werden deutlich umfangreicher als normale Hausaufgaben sein und dienen nicht zur Verbesserung der Note.
Zusatzaufgaben müssen mit **80%** bestanden werden, um wie folgt im Durchschnitt der n-2 besten Hausaufgaben berücksichtigt zu werden:
Wenn bei einer Zusatzaufgabe **100%** der Punkte erreicht wurden, wird sie wie eine **40%** Hausaufgabe bei der Berechnung berücksichtigt. Bei **90%** wird sie wie eine **30%** Aufgabe bewertet, bei **80%** wird sie wie eine **20%** Aufgabe bewertet.

Die Veranstaltung gilt als nicht bestanden, wenn mehr als **zwei Hausaufgaben nicht abgegeben** wurden oder der Mittelwert der zur Benotung herangezogenen Abgaben **weniger als 50%** beträgt.

Fragen bitte an hahn@uni-kassel.de

Aufgabe 1 Command Pattern

In dieser Aufgabe soll Bomberman als rundenbasiertes Singleplayerspiel mit KI unter Verwendung des Command Patterns realisiert werden.

- Erstellt ein Kassendiagram (SDMlib, xcore, emf, UMLLab,…) (das Datenmodell soll nicht per Hand geschrieben sein)
- Baut die Strategien aus Hausaufgabe 3 (moveUp, moveDown, moveLeft, moveRight, blast und stay) zu Commands um. Es muss für jedes Command eine do() und undo() Funktion geben.
- Fügt in euer Datenmodel ein BombermanKiPlayer hinzu, der automatisch das nächste Command auswählt. Die Auswahl kann zufällig geschehen. Es muss aber darauf geachtet werden, dass sich die KI nicht selber „sprengt“.
- Eine Runde ist zu Ende, wenn jede KI und der Spieler ein Command ausgewählt haben.
- Nach jeder Runde soll bei allen Bomb Objekten das blastTimeOut-Attribute um eins verringert werden. Wenn der Wert 0 erreicht, sollen im range Bereich alle Spieler entfernt und weitere Bomben „gezündet“ werden. Gezündet heißt, blastTimeOut-Attribute auf 0 setzen usw. (hier soll auch das Command Pattern verwendet werden)
- Das aktuelle Spielgeschehen muss auf einer Oberfläche geeignet angezeigt werden. (es kann die Oberfläche aus Hausaufgabe 5 wiederverwendet werden)
- Der Spieler soll in die Möglichkeit haben, das aktuelle Spielgeschehen Mittels Undo- und Redo-Funktionalität rundenweise vor und zurück zu stellen.
- Schreibt einen Junit-Test in welchem
    - eine sinnvolle Objektstruktur erzeugt wird
    - sinnvolle Assertions verwendet werden
    - einige Spielschritte durchgespielt werden
    - die Undo- und Redo- Funktionalität überprüft
    - die korrekte Anzeige der Spielsituation muss (stichprobenartig) geprüft werden