

Diese Aufgabe ist zu bearbeiten wenn Sie über eine **ungerade** Matrikelnummer verfügen.

Die Aufgaben müssen einzeln bearbeitet und abgegeben werden. Die Abgabe muss bis **spätestens Donnerstag 06.12.2011 um 23:59 Uhr** über unser Hausaufgabenabgabesystem <http://seblog.cs.uni-kassel.de/pmws1213/> erfolgen. Die Abgabe ist nur als einzelne *.zip oder *.jar-Datei möglich. Daher müssen alle für eine Abgabe relevanten Daten zu einem solchen Archiv kombiniert werden. Diese Hausaufgabe gibt **12 Punkte**.

Hinweise zur Abgabe:

- Die Hausaufgabe als exportiertes Eclipse Projekt (*.zip, **nicht** den gesamten Workspace) abgeben. Das kann mit Hilfe der Eclipse Export Funktion durchgeführt werden. Ist das 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 und alle zukünftigen Abgaben nach folgendem Schema:

PMWS1112_HA<a>_<Matrikelnummer>,

wobei <a> für die Nummer der aktuellen Hausaufgabe steht. Beispiel:

PMWS1112_HA5_12345679.

Vorbereitung

Für die Bearbeitung der Hausaufgabe 5 benötigen Sie:

- Ein aktuelles *Fujaba4Eclipse*-PlugIn für Eclipse. Installieren bzw. aktualisieren Sie das PlugIn über den *Install new software* Dialog bzw. über den Eclipse Update Mechanismus. Die URL für die Update-Site lautet: <http://www.se.eecs.uni-kassel.de/fileadmin/se/update>. Das Vorgehen kann mit Hilfe des Screencasts der Übungen nachvollzogen werden.
- Die Implementierung des Klassendiagramms von Diaballik. Sie können eine fertige Version vom Blog zur Veranstaltung herunterladen (<http://seblog.cs.uni-kassel.de/category/currentterm/pm-ws1213/>) und als Projekt in Eclipse importieren.

Aufgabe 1 - Storyboards (12P)

Erstellen Sie Fujaba4Eclipse Storyboards zu folgenden Szenarien. Für jedes Szenario werden 3 Punkte vergeben. Passen Sie die Methoden `DiaballikGame::checkEnd()`, `Holder::move(Field f)` und `Ball::pass(Holder h)` soweit an, bis alle von den Storyboards generierten JUnit Tests erfolgreich durchlaufen. Die Endsituation und dementsprechend die Implementierung der drei Methoden müssen die offiziellen PMWS12/13 Diaballik Regeln beachten (zu finden im PMWS12/13 Blog!).

Randbedingungen:

- 1 Spiel mit Namen "diaballik"
- 2 Spieler, Alice (rot) und Bob (grün)
- In der Startsituation müssen zusätzlich lediglich die gelb umrandeten Informationen modelliert werden. Alle nicht benannten Objekte dürfen beliebige Objektnamen haben.

Abzugeben ist das exportierte Eclipse Projekt, welches ein Fujaba Projekt mit den vier Storyboards sowie die korrekte und allgemein gültige Implementierung der drei obigen Methoden enthält.

- *Roter Ball in grüner Endzone*

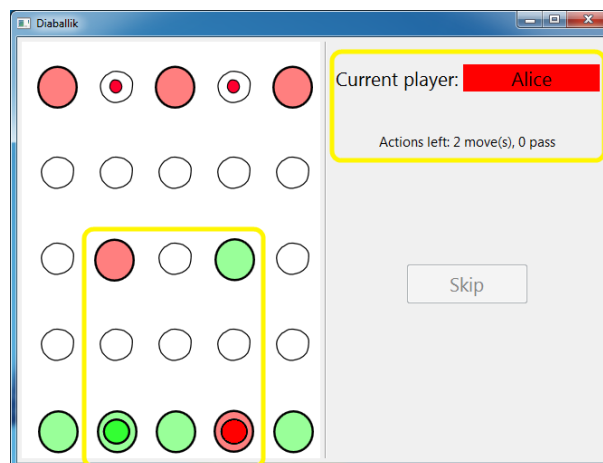


Abbildung 1: Roter Ball in grüner Endzone

Der rote Spieler Alice hat bereits einen Pass ausgeführt. Aufruf von `DiaballikGame::checkEnd()`.

- *Horizontales Bewegen eines Spielsteins*

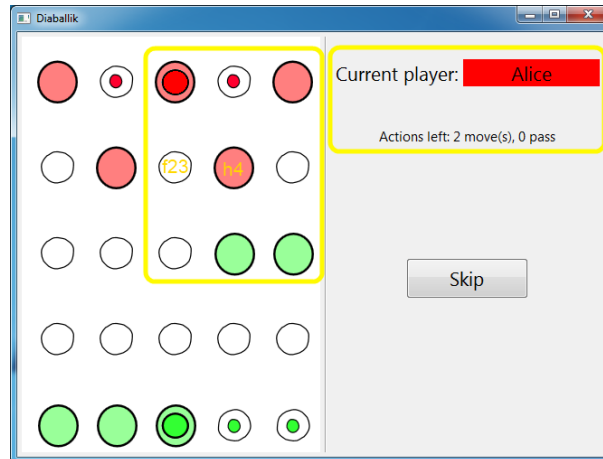


Abbildung 2: Horizontales Bewegen eines Spielsteins

Der rote Spieler Alice ist an der Reihe. Aufruf von `Holder::move(f23)` auf Holder `h4`.

- *Diagonaler Pass*

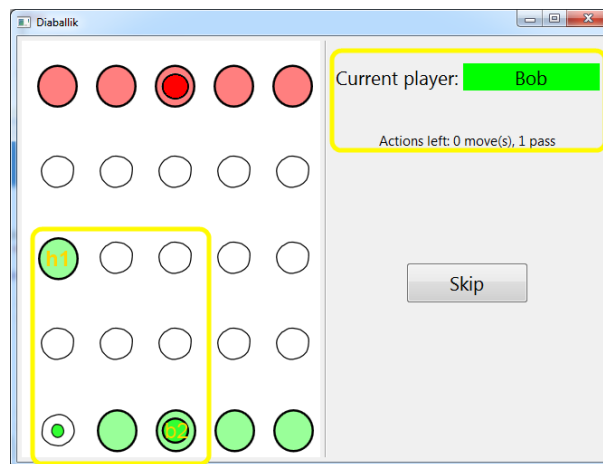


Abbildung 3: Diagonaler Pass

Der grüne Spieler Bob ist an der Reihe. Aufruf von `Ball::pass(h1)` auf Ball `b2`.

- Vertikaler Pass über einen eigenen Spielstein

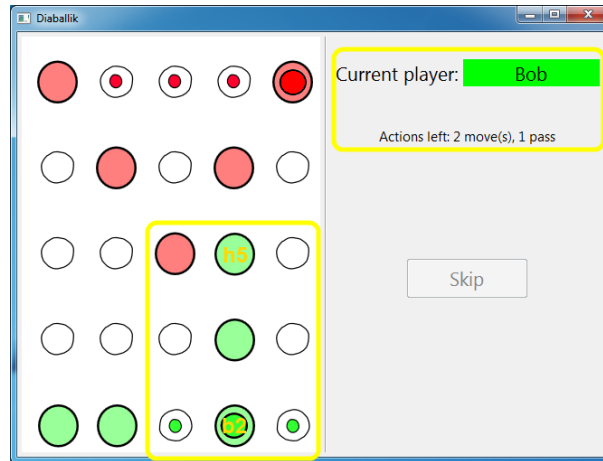


Abbildung 4: Vertikaler Pass über einen eigenen Spielstein

Der grüne Spieler Bob ist an der Reihe. Aufruf von `Ball1::pass(h5)` auf Ball b2.