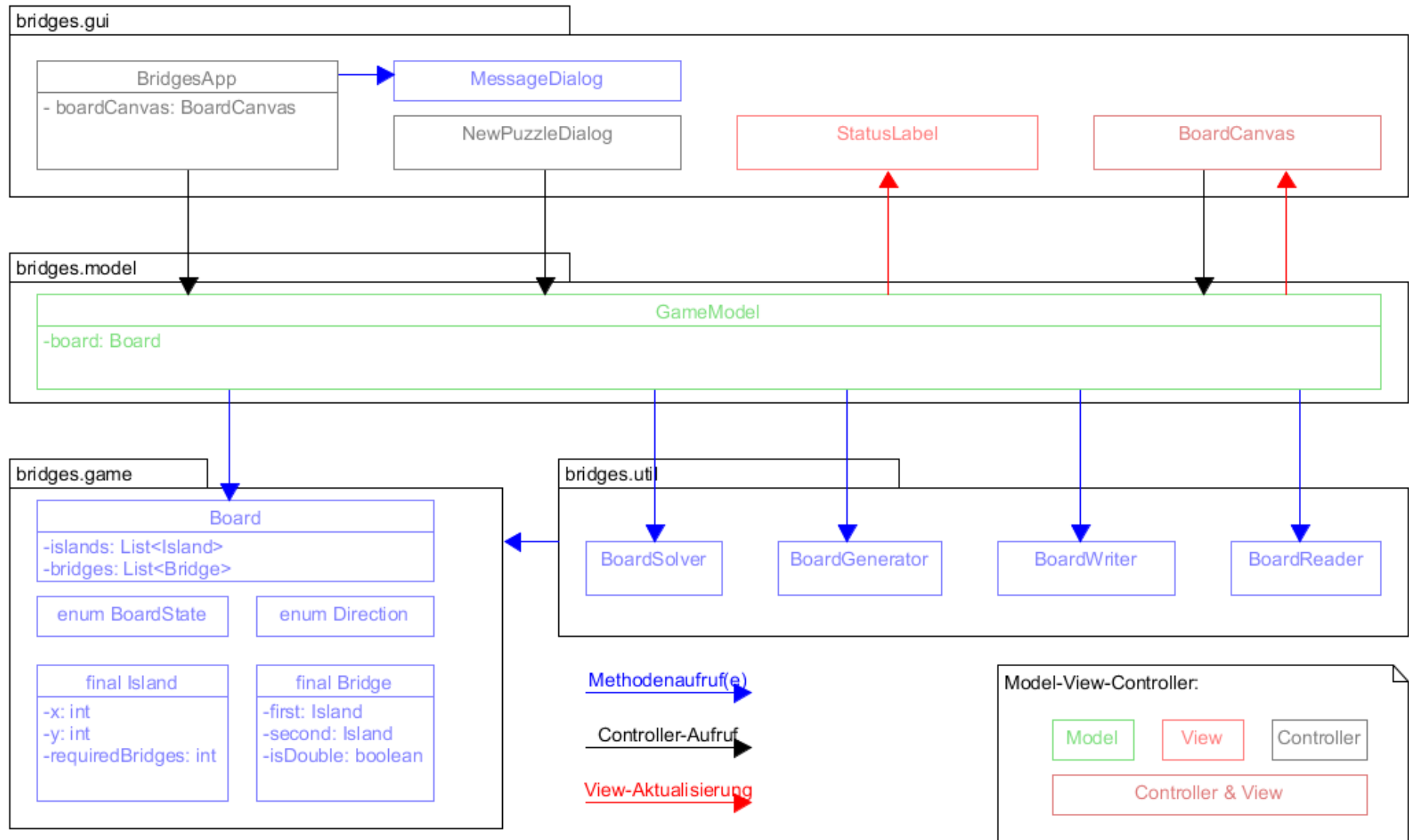


Architekturbeschreibung: Bridges | Maik Messerschmidt (9440577)*



* Dargestellt sind die wichtigsten Klassen des Projektes mit bedeutenden Membervariablen sowie eine einfache Veranschaulichung von Aufrufen der Klassen untereinander und die Bedeutung der Klassen in ihrer Rolle als Model (Modell), Präsentation (View) oder Steuerung (Controller).

Das Paket **bridges.game** stellt grundlegende Spielfunktionalität zur Verfügung. Die Klassen `Island` und `Bridge` sind dabei unveränderliche Objekte, so dass diese von einem `Board`-Objekt frei an andere Objekte weitergereicht werden können, ohne eine Zustandsänderung des `Board`-Objekts zu ermöglichen.

Das Paket **bridges.util** stellt Helferklassen für das Lesen, Schreiben, Lösen und Generieren von Spielen dar. Die meisten dieser Klassen werden nur statisch benutzt.

Das Paket **bridges.model** besitzt mit der Klasse `GameModel` ein thread-sicheres Interface, dass z.B. in einer Modell-Präsentation-Steuerung-Architektur mit mehreren Steuerungen eingesetzt werden kann. Es besitzt eine `Board`-Instanz, die transparent durch Laden oder Speichern durch eine andere ersetzt werden kann. Durch verschiedene Methoden wird die benötigte Funktionalität der Pakete `bridges.game` und `bridges.util` zur Verfügung gestellt.

Das Paket **bridges.gui** schließlich realisiert die grafische Benutzeroberfläche. Dabei wird ein Großteil des Hauptfensters durch die Klasse `BridgesApp` implementiert. Lediglich das Spielfeld wird durch die Klasse `BoardCanvas` ermöglicht. `NewPuzzleDialog` ist eine weitere bedeutende Klasse, welche den Dialog für neue Puzzle realisiert. Fehlermeldungen werden der Benutzerin oder dem Benutzer durch die Klasse `MessageDialog` angezeigt. Alle Klassen bis auf `BoardCanvas` haben dabei in der Modell-Präsentation-Steuerung-Architektur nur die Rolle von Steuerung oder Präsentation, während `BoardCanvas` beide Rollen einnimmt.