

```

        });
    });
}
...
})();

```

Jak widać, wywołanie `drawJewel()` w metodzie `render()` ma teraz dodatkowe — czwarty i piąty — argumenty. Argumenty te oznaczają skalę i stopień obrotu, które należy nałożyć na klejnot przed jego namalowaniem. Listing 9.14 przedstawia zmiany w funkcji `drawJewel()` w skrypcie `display.canvas.js`.

Listing 9.14. Skalowanie i obrót w metodzie `drawJewel()`

```

jewel.display = (function() {
    ...
    function drawJewel(type, x, y, scale, rot) {
        var image = jewel.images["images/jewels" +
            jewelSize + ".png"];
        ctx.save();
        if (typeof scale !== "undefined" && scale > 0) {
            ctx.beginPath();
            ctx.rect(x,y,1,1);
            ctx.clip();
            ctx.translate(x + 0.5, y + 0.5);
            ctx.scale(scale, scale);
            if (rot) {
                ctx.rotate(rot);
            }
            ctx.translate(-x - 0.5, -y - 0.5);
        }
        ctx.drawImage(image,
            type * jewelSize, 0, jewelSize, jewelSize,
            x, y, 1, 1
        );
        ctx.restore();
    }
    ...
})();

```

Jeżeli przypominasz sobie, o czym pisałem w rozdziale 6., dodane transformacje nie powinny Cię przerażać. W przypadku funkcji `removeJewel()` argument skalowania jest przekazywany w postaci wyrażenia `(1-pos)`, którego wartość wynosi od 1 do 0 w różnych etapach animacji. Wartość `pos` znajdziesz również w argumentcie `rot`, z tym że tam jest ona mnożona przez π , aby otrzymać pełen obrót o 360 stopni.

Wskazówka: Dzięki naszym zabiegom możesz łatwo dodawać współczynnik skalowania. Spróbuj więc nadać mu wartość 1.1 i użyć w funkcji `renderCursor()`. W rezultacie otrzymasz wyraźniejszą poświatę.

Ponowne wypełnianie planszy

Gdy moduł planszy wykryje, że gracz nie może wykonać żadnego ruchu, wywoła zdarzenie `refill`. Argument przesłany wraz ze zdarzeniem zawiera klejnoty, które mają się pojawić na nowej planszy.