



Rysunek 9.4. Po dodaniu obsługi `mousePressed()` oraz ustawieniu `BubbleListener` w roli interfejsu słuchacza zdarzeń myszy w aplikacji możemy rysować bąbelki, klikając dowolne miejsce na ekranie

Odbieranie zdarzeń ruchu myszą

Zdarzenia ruchu myszą różnią się od zdarzeń naciśnięcia przycisku myszy, lecz mimo to mogą być przetwarzane w klasie rozszerzającej klasę `MouseListener`, takiej jak `BubbleListener`. Istnieją dwa typy zdarzeń ruchu myszą: `mouseMoved()` oraz `mouseDragged()`.

Zdarzenie `mouseMoved()` następuje, gdy wskaźnik myszy przemieszcza się nad kanwą rysunku. Natomiast zdarzenie `mouseDragged()` następuje, gdy wskaźnik myszy przemieszcza się i jednocześnie naciśnięty jest przycisk myszy. Ponieważ chcemy rysować bąbelki tylko wtedy, gdy użytkownik klika i przeciąga wskaźnik myszy, zaprogramujemy najpierw obsługę zdarzenia `mouseDragged()` w klasie `BubbleListener`, a następnie dodamy do konstruktora `BubblePanel` interfejsu słuchacza ruchu, aby aktywować działanie słuchacza.

Dodajemy metodę obsługi zdarzeń `mouseDragged()` do klasy `BubbleListener`, jak pokazano poniżej:

```
private class BubbleListener extends MouseAdapter {
    public void mousePressed(MouseEvent e) {
        --wycinek--
    }
    public void mouseDragged(MouseEvent e) {
        bubbleList.add(new Bubble(e.getX(), e.getY(), size));
        repaint();
    }
}
```

Jak widać, kod wygląda niemal tak samo jak kod metody obsługi zdarzenia `mousePressed()`, różni się jedynie nazwą `mouseDragged()`. Wynika to z faktu, iż obie metody obsługują zdarzenia `MouseEvent`. Metoda `mousePressed()` obsługuje zdarzenia, gdy użytkownik naciśnie przycisk myszy, natomiast metoda `mouseDragged()` jest wywoływana, gdy użytkownik przeciągnie wskaźnik myszy. W tej aplikacji oba zdarzenia powinny dawać ten sam rezultat – dodanie bąbelka `bubbleList` i wywołanie funkcji `repaint()`.

A teraz dodajemy `BubbleListener` jako słuchacza zdarzeń ruchu myszy do konstruktora `BubblePanel()`, używając następującej instrukcji:

```
public BubblePanel() {
    bubbleList = new ArrayList<Bubble>();
    setBackground(Color.BLACK);
    // testBubbles();
    addMouseListener( new BubbleListener() );
    addMouseMotionListener( new BubbleListener() );
}
```

Ponownie uruchamiamy aplikację w celu przetestowania tego nowego słuchacza zdarzeń. Zapisujemy kod, naciskamy przycisk uruchamiania i klikamy oraz przeciągamy wskaźnik myszy w wyświetlonym oknie aplikacji, jak pokazano na rysunku 9.5.



Rysunek 9.5. Teraz możemy rysować ciągłe strumienie bąbelków, przeciągając wskaźnik myszy

Imponujące, prawda? W języku Java wystarczy dodać do `BubblePanel` dwie metody obsługi zdarzeń przycisku oraz ruchu wskaźnika myszy, aby uzyskać interaktywną, kolorową aplikację rysującą.