

Hello Wien – rajzoló része

Most csak a rajzolást emelem ki.

A tényleges rajzolást a CsataPanel paintComponent() metódusa végzi, de a vezérlő osztály rajzolas() metódusa mondja meg, hogy mit rajzoljon.

A panel rajzoló metódusa:

```
private Image kep = new ImageIcon(this.getClass().
    getResource("/adatok/var9.png")).getImage();

@Override
protected void paintComponent(Graphics g) {
    super.paintComponent(g);
    int kezdox = 0, kezdoy = 0,
        szelesseg = this.getWidth(),
        magassag = this.getHeight();
    g.drawImage(kep, kezdox, kezdoy, szelesseg, magassag, null);

    if(vezerlo != null){
        vezerlo.rajzolas(g);
    }
}
```

A Vezerlo osztályban:

```
/**
 * Megmondja, hogy ezeket a katonákat kell kirajzolni, azaz meghívja
 * a katona példány rajzolas() metódusát.
 *
 * @param g
 */
public void rajzolas(Graphics g) {
    for (Katona katona : harcosok) {
        katona.rajzolas(g);
    }
}
```

A Katona osztály rajzolas() metódusa:

```
public void rajzolas(Graphics g){
    g.setColor(szin);
    g.fillOval(kx-sugar, ky-sugar, 2*sugar, 2*sugar);
}
```

Azt, hogy hova kerüljenek a kiválasztott harcosok, a `Vezerlo` osztály állítja be. Jelenleg a csatamező véletlen pontjára, azaz vízszintesen a panel szélétben bárhova, függőlegesen csak az alsó valahányad részébe. A harcosok elhelyezése:

```
/**
 * Csataba küldi a kiválasztott katonákat, azaz mindegyiknek megmondja,
 * hogy hova álljon - ez a csatater egy véletlen pontja.
 *
 * @param valasztottKatonak
 */
public void csatabaAllit(List<Katona> valasztottKatonak) {
    this.harcosok = valasztottKatonak;
    for (Katona katona : harcosok) {
        katona.setKx((int) (Math.random() * (csataPanel.getWidth()
            - 2 * Katona.getSugar()) + Katona.getSugar()));
        katona.setKy((int) (Math.random() * (csataPanel.getHeight()
            - rajzMagassag - Katona.getSugar()) + rajzMagassag));
    }
    // frissítjük a csatapanelt
    csataPanel.repaint();
}
```

Azt, hogy egy adott pontban eltalálták-e valamelyik katonát, szintén a `Vezerlo` osztály tudja végigkérdezni:

```
/**
 * Mi történjen, ha a panel jelzi, hogy rákattintottak az adott helyen.
 * Végignézi az összes harcost, és ha valakit eltaláltak, akkor a megadott
 * eséllyel vagy meglövik, vagy ő lőtt. Kérjük a harcosok panelt, hogy az
 * eredmény alapján módosítsa a listafelületet. Ha már nem él, akkor
 * kivesszük a harcosok közül.
 *
 * @param x
 * @param y
 */
public void kattintottak(int x, int y) {
    for (Katona katona : harcosok) {
        if (katona.eltalaltak(x, y)) {
            if (Math.random() < SZAZALEK) {
                katona.megLottek();
            } else {
                katona.lott();
            }
            harcosokPanel.listaModositas(katona);
            if (!katona.isElo()) {
                harcosok.remove(katona);
            }

            // frissítjük a csatapanelt
            csataPanel.repaint();

            // Nagyon fontos ez a break, mert különben törlés után
            // összeomlik a ciklus.
            break;
        }
    }
}
```

de hogy tényleg eltaláltak-e, azt a katona tudja, mégpedig onnan, hogy a középpontja és az adott koordinátájú pont közötti távolság kisebb-e a kör sugaránál, vagy nem.

A Katona osztályban:

```
public boolean eltalaltak(int x, int y) {  
    double tav = Math.sqrt((x-kx)*(x-kx) + (y-ky)*(y-ky));  
    return (tav < sugar);  
}
```

Azt pedig, hogy a felhasználó kattintott-e, a CsataPanel osztály tudja, vagyis itt hívjuk meg a kattintottak() metódust:

```
private void formMousePressed(java.awt.event.MouseEvent evt) {  
    vezerlo.kattintottak(evt.getX(), evt.getY());  
}
```