Progress Bar and Countdown Timer Features - Hud Class

Overview

The `Hud` class has been extended with two new methods:

1. `drawProgressBar` - Displays visual progress bars
2. `drawCountdown` - Displays countdown timers with text or graphical representation

---

Progress Bar Method

Method Signature

```
public void drawProgressBar(Graphics2D g, int row, String label, double progress,
int barWidth)
```

Parameters

- `g` (Graphics2D): The graphics context to draw on
- `row` (int): The row number where the progress bar should be drawn (follows the same row system as text lines)
- `label` (String): The text label to display before the progress bar
- `progress` (double): The progress value, ranging from 0.0 (0%) to 1.0 (100%)
- `barWidth` (int): The width of the progress bar in pixels (e.g., 200)

Features

1. **Color-coded Progress**: The bar automatically changes color based on progress:
    - **Red**: progress < 33%
    - **Yellow**: 33% ≤ progress < 66%
    - **Green**: progress ≥ 66%
2. **Percentage Display**: Shows the progress percentage as text next to the bar
3. **Automatic Clamping**: Progress values are automatically clamped to the valid range [0.0, 1.0]
4. **Consistent Styling**: Integrates seamlessly with the existing HUD system (font, positioning, colors)

Countdown Timer Method

Method Signature

```
public void drawCountdown(Graphics2D g, int row, String label, double
remainingSeconds, double totalSeconds, boolean graphical)
```

Parameters

- `g` (Graphics2D): The graphics context to draw on
- `row` (int): The row number where the countdown should be drawn
- `label` (String): The text label to display before the countdown
- `remainingSeconds` (double): The remaining time in seconds
- `totalSeconds` (double): The total countdown duration in seconds (used for graphical representation)
- `graphical` (boolean): If true, shows a visual bar; if false, shows only text

Features

1. **Flexible Time Format**:
    - Times ≥ 60 seconds: Displays as "M:SS" (e.g., "2:30")
    - Times < 60 seconds: Displays as "SS.S" (e.g., "45.3s")
2. **Text Mode**: Simple text display of remaining time
3. **Graphical Mode**: Visual countdown bar with color coding:
    - **Green**: More than 66% of time remaining
    - **Yellow**: 33% to 66% of time remaining
    - **Red**: Less than 33% of time remaining (urgent!)
4. **Automatic Clamping**: Remaining seconds are clamped to prevent negative values
5. **Perfect for Weapon Reloads**: Ideal for showing ammunition reload timers

---

Usage Examples

Progress Bar - Basic Usage

```
// Create a HUD instance
Hud hud = new Hud(Color.GRAY, 10, 12, 35);
hud.maxLenLabel = 7; // Set label width for proper alignment

// In your rendering method
public void render(Graphics2D g) {
    // Draw a health bar at 75% (green)
    hud.drawProgressBar(g, 1, "Health", 0.75, 200);

    // Draw an energy bar at 50% (yellow)
    hud.drawProgressBar(g, 2, "Energy", 0.50, 200);

    // Draw a shield bar at 25% (red)
```

```
    hud.drawProgressBar(g, 3, "Shield", 0.25, 200);
}
```

Countdown Timer - Weapon Reload Example

```
// Create a HUD instance
Hud hud = new Hud(Color.GRAY, 10, 12, 35);
hud.maxLenLabel = 10; // Set label width for weapon names

// In your rendering method
public void render(Graphics2D g) {
    double weapon1Reload = 5.3;  // 5.3 seconds remaining
    double weapon1Total = 10.0;  // 10 seconds total reload time

    double weapon2Reload = 125.0; // 2 minutes 5 seconds remaining
    double weapon2Total = 180.0;  // 3 minutes total reload time

    // Draw graphical countdown bars
    hud.drawCountdown(g, 1, "Rifle", weapon1Reload, weapon1Total, true);
    hud.drawCountdown(g, 2, "Launcher", weapon2Reload, weapon2Total, true);

    // Or draw text-only countdown
    hud.drawCountdown(g, 3, "Pistol", 3.5, 0, false);
}
```

Using the Example Classes

The `ProgressBarHud` and `CountdownHud` classes provide ready-to-use examples:

```
// Progress bars
ProgressBarHud progressHud = new ProgressBarHud();
progressHud.drawWithProgressBars(g, 0.85, 0.40, 0.95);

// Weapon reload timers (graphical)
CountdownHud countdownHud = new CountdownHud();
countdownHud.drawWeaponReloads(g, 5.3, 10.0, 125.0, 180.0);

// Weapon reload timers (text only)
countdownHud.drawWeaponReloadsTextOnly(g, 5.3, 125.0);
```

Custom HUD with Mixed Features

You can combine both features in a custom HUD class:

```
public class MyGameHud extends Hud {
    public MyGameHud() {
        super(Color.CYAN, 10, 12, 35);
```

```
        this.maxLenLabel = 12;
    }

    public void drawGameStats(Graphics2D g, PlayerStats stats, WeaponStats weapon)
{
        // Show player stats as progress bars
        drawProgressBar(g, 1, "Health", stats.getHealthRatio(), 200);
        drawProgressBar(g, 2, "Shield", stats.getShieldRatio(), 200);

        // Show weapon reload as countdown
        if (weapon.isReloading()) {
            drawCountdown(g, 3, "Reloading",
                          weapon.getReloadTimeRemaining(),
                          weapon.getTotalReloadTime(),
                          true);
        }
    }
}
```

Visual Appearance

Progress Bar

```
Health  [████████████▒▒▒▒▒▒▒▒] 60%
Energy  [██████▒▒▒▒▒▒▒▒▒▒▒▒▒▒] 30%
Shield  [████████████████████] 100%
```

Countdown Timer (Graphical)

```
Rifle     [████████▒▒▒▒▒▒▒▒▒▒▒▒] 5.3s
Launcher  [████████████▒▒▒▒▒▒▒▒] 2:05
```

Countdown Timer (Text Only)

```
Pistol    3.5s
Sniper    1:45
```

Notes

- Both methods automatically calculate dimensions based on the font size
- Both methods are vertically aligned with the text baseline
- Graphics state (colors) is automatically preserved and restored after drawing
- The methods work with any existing HUD positioning configuration

- **Countdown bars** show remaining time (bar fills from full and decreases as time runs out)
  - Full bar (green) = plenty of time remaining
  - Partial bar (yellow) = time getting low
  - Almost empty (red) = urgent, almost out of time
- **Progress bars** show completion (bar fills from empty and increases as progress advances)
  - Empty bar (red) = just started
  - Partial bar (yellow) = making progress
  - Full bar (green) = nearly complete