### Barbarian's uprise factor is almost always 1 or 2

**Description**

If barbarian's target nation has 10 to 111 cities, uprise factor is 2. (if target has 112 to 1253 cities, factor is 3)

It is ok?

I think this code:

```c
city_max *= 1.2 + UPRISE_CIV_SIZE;
```

is

```c
city_max = city_max * 6 / 5 + UPRISE_CIV_SIZE;
```

(server/barbarian.c try_summon_barbarians)

(Actually, if civ has 111 cities, uprise factor seems 3...maybe floating point arithmetic error?)

e.g. If barbarians = hordes and you have 110 cities, barbarian comes with 8 to 10 units + leader.

If you have 111 cities, barbarian comes with 12 to 14 units + leader.

### History

**#1 - 2019-12-18 12:29 PM - Jacob Nevins**

- Sprint/Milestone changed from 2.6.1 to 2.6.2

**#2 - 2019-12-22 08:01 AM - Marko Lindqvist**

Akechi . wrote:

If barbarian's target nation has 10 to 111 cities, uprise factor is 2. (if target has 112 to 1253 cities, factor is 3)

It is ok?

I think this code:

```c
city_max *= 1.2 + UPRISE_CIV_SIZE;
```

is

```c
city_max = city_max * 6 / 5 + UPRISE_CIV_SIZE;
```

(server/barbarian.c try_summon_barbarians)

No, it's

```c
city_max = city_max * ( 6 / 5 + UPRISE_CIV_SIZE );
```

**#3 - 2019-12-22 05:52 PM - Akechi .**

Marko Lindqvist wrote:

No, it's

```c
city_max = city_max * ( 6 / 5 + UPRISE_CIV_SIZE );
```

That means uprise factor is intended to set 1 or 2 in general, not 3 or higher?

**#4 - 2020-02-01 03:28 PM - Jacob Nevins**

- Subject changed from Barbarian's uprise factor is almost 1 or 2 to Barbarian's uprise factor is almost always 1 or 2

Previous comments hinge on C semantics of `*=`. Just to confirm, C90 6.3.16.2 says:

A compound assignment of the form E1 op= E2 differs from the simple assignment expression E1 = E1 op (E2) only in that the lvalue E1 is evaluated only once.

So behaviour is as cazfi describes.

Behaviour more like akechi describes would be given by `city_max = city_max * 1.2 + UPRISE_CIV_SIZE`. 
Here's a comparison of what I think the results are (not sure why they don't quite match akechi's figures):

<table>
<thead>
<tr>
<th>uprise</th>
<th>current</th>
<th>akechi</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&lt;10 cities</td>
<td>&lt;10 cities</td>
</tr>
<tr>
<td>2</td>
<td>&lt;110 cities</td>
<td>&lt;22 cities</td>
</tr>
<tr>
<td>3</td>
<td>&lt;1210 cities</td>
<td>&lt;36 cities</td>
</tr>
<tr>
<td>4</td>
<td>&lt;13310 cities</td>
<td>&lt;53 cities</td>
</tr>
</tbody>
</table>

This code was written relatively recently, by cazfi in 2015 ([gna patch #5909](https://gna.org/); and the preceding code was too mad to discern intent. So we can hope to learn the original intent of this code. cazfi, do you remember? (Ticket says "This patch makes barbarian gang size to steadily grow as player gets bigger", hinting that current behaviour is not what was intended.)

If the current behaviour isn't what was intended, it seems a bit late to change on a stable branch (the current behaviour reportedly had lots of playtesting even before commit). Maybe fix for 3.0?

#5 - 2020-02-01 04:48 PM - Marko Lindqvist

I'm guessing my commit message is what's wrong. Logarithmic scale sounds correct, as individual cities (that barbarians will attack) are not going stronger as fast as number of cities in the nation grows. We don't want to punish fast expansion too much. However, while the shape of the curve is right, maybe the numbers are a bit high. Should we adjust UPRISE_CIV_SIZE a bit?

#6 - 2020-02-09 03:16 PM - Jacob Nevins

- Sprint/Milestone changed from 2.6.2 to 2.6.3

#7 - 2020-03-01 04:54 AM - Akechi .

Marko Lindqvist wrote:

I'm guessing my commit message is what's wrong. Logarithmic scale sounds correct, as individual cities (that barbarians will attack) are not going stronger as fast as number of cities in the nation grows. We don't want to punish fast expansion too much. However, while the shape of the curve is right, maybe the numbers are a bit high. Should we adjust UPRISE_CIV_SIZE a bit?

I tried making some formulas... how about these?

```c
#define UPRISE_CIV_SIZE   8 and next city_max += uprise * 4 + 2;
```

or

```c
#define UPRISE_CIV_SIZE   6 and next city_max += uprise * 3 + 2 - (uprise % 2);
```

or

```c
#define UPRISE_CIV_SIZE   4 and next city_max += uprise * 2 + 1;
```

I tried making some formulas... how about these?

### number of victim player's city needed

<table>
<thead>
<tr>
<th>uprise</th>
<th><em>11.2 <em>1.2+10 +uprise</em>4+2 +uprise</em>3+2-(uprise%2) +uprise*2+1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0+ 0+ 0+ 0+</td>
</tr>
<tr>
<td>2</td>
<td>10+ 10+ 8+ 6+ 4+ (UPRISE_CIV_SIZE)</td>
</tr>
<tr>
<td>3</td>
<td>112+ 22+ 18+ 14+ 9+</td>
</tr>
<tr>
<td>4</td>
<td>1254+ 36+ 32+ 24+ 16+</td>
</tr>
<tr>
<td>5</td>
<td>14044+ 53+ 50+ 38+ 25+</td>
</tr>
<tr>
<td>6</td>
<td>157292+ 73+ 72+ 54+ 36+</td>
</tr>
<tr>
<td>7 can't reach</td>
<td>97+ 98+ 74+ 49+</td>
</tr>
<tr>
<td>8 can't reach</td>
<td>126+ 128+ 96+ 64+</td>
</tr>
<tr>
<td>9 can't reach</td>
<td>161+ 162+ 122+ 81+</td>
</tr>
<tr>
<td>10 can't reach</td>
<td>203+ 200+ 150+ 100+</td>
</tr>
<tr>
<td>11 can't reach</td>
<td>253+ 242+ 182+ 121+</td>
</tr>
<tr>
<td>12 can't reach</td>
<td>313+ 288+ 216+ 144+</td>
</tr>
</tbody>
</table>

#8 - 2020-12-31 05:30 AM - Marko Lindqvist

- Sprint/Milestone changed from 2.6.3 to 2.6.4

#9 - 2021-01-03 10:14 PM - Marko Lindqvist

- File 0033-Adjust-formula-for-barbarian-band-size.patch added

- Status changed from New to Resolved

How about this patch?

It should produce

Player cities -> band size factor:

| ...10 | -> | 1 |
| 11...31 | -> | 2 |
#10 - 2021-01-08 07:36 PM - Marko Lindqvist
- Status changed from Resolved to Closed
- Assignee set to Marko Lindqvist

Files

<table>
<thead>
<tr>
<th>File Name</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>0033-Adjust-formula-for-barbarian-band-size.patch</td>
<td>1016 Bytes</td>
<td>2021-01-03</td>
<td>Marko Lindqvist</td>
</tr>
</tbody>
</table>