

## New in Version 1.3

---

EveryGame 1.3 provides many new features for supplemental boards. The name specified for boards past the first will now be listed on their tabs. In addition the board attributes "edge" and "position" allow you to control tab placement. For instance:

```
<board edge="3" position="0.5">Extra Cards/board</board>
```

Will create a tab called "Extra Cards", half way along the screen edge with the Options tab.

The <location\_display> tag has also been updated, with options to specify the font size and color. In addition, empty locations may have their text display disabled, and new display types that list either the location's name, or the name of it's top-most piece have been added.

```
<location_display type="piece" show_empty="0" font_size="12" font_color="0,255,0">
```

Will have the location display the name of the piece it contains (when not empty), in green, 12 point text.

The piece action "random\_side\_at\_location" has been added. This allows you to create a button that would roll whatever dice are in specified locations, allowing for dice-locking by pulling them in / out of the randomized location. You would use it like this:

```
<single_tap action="random_side_at_location" args="location_name,0_1_2_3_4_5">
```

EveryGame can now natively handle .zip files (as well as .zip files renamed to .ege). The .zip file should contain the game directory(s) at its top level. As an example, if you had a "Checkers" directory, that contained all of your "Checkers\_\*" files, you could right/ctrl click on the "Checkers" directory, and select "Compress". EveryGame has registered these file types with iOS, which means that games can be copied to EveryGame without iTunes. Instead, you can tap a download link in Safari, or tap and hold on an email attachment, and choose to open them in EveryGame. You can also copy them to EveryGame via the normal iTunes method. Currently, the in-game web browser does not support opening some .zip files (off of the Google Group for instance), though links can be copied, and pasted into Safari to open them.

Further, additional bug fixes are implemented, among which the game will automatically save when restarted, and EveryGame will not crash if a piece's initial location does not exist. An issue with using the move\_piece action to remove pieces from stacks was corrected, as was the lack of 6's on the Backgammon dice. Images are now viewable after a game is copied from the iPad to the computer via iTunes.

## New in Version 1.2

---

EveryGame version 1.2 focuses on adding tools to make debugging easier. The "Create a New Game" menu has expanded help, and also links to various online resources and forums. There is also an "Edit Game" menu available under the Game Options tab, that reports on detected errors, and allows for xml files to be edited directly on the iPad. Missing images are also better detected.

The engine has been enhanced to allow new options for xml files, and for common use-cases to be expressed more compactly.

```
<piece count="8">tree</piece>
```

The piece tag now supports a count argument. For example, the code above would create 8 trees (named tree\_0 through tree\_8) all with the same parameters and location.

```
<location rows="7" cols="7">square</location>
```

Similarly, locations now have new fields which let you specify entire grids at once. The example here creates a 7x7 grid, named "square\_0\_0" through "square\_6\_6"

```
<single_tap action="send_to_back">
```

We also now have a way to cycle through the pieces in a location. The "send\_to\_back" option takes a piece, and puts it at the bottom of whatever stack / queue contains it.

```
<single_tap action="move_piece_at_location" args="stock, square_0_0">
```

Finally, the "move\_piece\_at\_location" allows you to trigger a piece movement with taps. However, unlike the

"move\_piece" action that existed before, this lets you specify any location on the board to move the piece from.

### **New in Version 1.1**

---

EveryGame version 1.1 expands the engine to include some new features. Many of them are designed to aid the creation of "buttons", that perform many actions (possibly on other pieces) when tapped. For instance, this block of code highlights several of them:

```
<draggable value="0">
  <single_tap action="random_side" args="v2,1_2_3_4_5_6,dice1">
    <single_tap action="move_piece" args="stock,dice1">
      <piece>button</piece>
    </single_tap>
  </single_tap>
</draggable>
```

The piece named "button" will appear on the screen, and respond to taps, but will not respond to drags (though it can still be positioned via <initial\_location> or "move\_piece" actions). When tapped, it will both roll and move "dice1", but do nothing to itself. This opens up a lot of possibilities, like one button to roll multiple dice, etc.

The other new tags added are:

<capture\_location> and <bounce\_location>

these tags are placed onto pieces, and specify what should happen if a piece is moved onto an occupied cell. First, the moving piece's capture\_location is checked. If it exists, the piece being moved onto is moved there, freeing the square. If that is not possible, the game will attempt to move the stationary piece to its bounce\_location. If neither of these are possible, the move will not occur.