**Conventions Document**

**V1.0 - 03/02/2016**

Naming Game Assets

All assets should be give an appropriate prefix to allow differentiation between types in code. All assets should be stored in appropriately named folders, assets that are level dependent should be placed in a master folder named after each level. Folder names should capitalise the first letter of each word and may use spaces. Asset names should be snake case; in all lower case and should separate words with underscores. Names should be structured to describe the asset from generic terms and becoming more specific.

Prefix List:

* Sprites – spr\_
* Sounds – snd\_
* Backgrounds – bg\_
* Paths – pth\_
* Scripts – scr\_
* Shaders – shd\_
* Fonts – fnt\_
* Time Lines – tln\_
* Objects – obj\_
* Rooms – rm\_

For example, a sprite of the player’s right walk cycle should be stored in the “Player Animations” folder and should be named “spr\_player\_walk\_right”

Script Creation

Scripts should be created as separate script assets and called from the game objects, as opposed to defining them in the game objects, this encourages code re-use and helps to minimise commit conflicts.

Step Event

For efficiency purposes, use of the step event should be minimised. It is encouraged to call functions from alternate events wherever possible.

Code Conventions – Naming Variables

All variables should declared in snake case. GML’s requirement that global variables be accessed through the global container means they are already suitably differentiated. Local variables should be proceeded with “l\_”.

Code Conventions – Bracket Placement

Opening scope brackets should be placed on the same line as the scope header. Close scope braces should be placed on a new line.

Commit Conventions

Commit summaries should include a suitable description of what was edited/created in the commit. Blocks of work that have achieved multiple goals should be divided into multiple commits.