Wonkypix

presents

Config Driven 2D Game Engine
©2015 Jason Brooks

Author:	Jason Brooks
Engine Version:	1.6b
Document Version:	V0.2 Draft
Published Date:	TBC
Last Reviewed Date:	7th May 2020 – Bob McFarlane
Artwork ©	Andy Noble
Additional Artwork ©	Nathan Smith
Music ©	Andy Noble
Game Design ©	Matthew Smith
Coding ©	Jason Brooks

1 2D Platform Gaming Engine Manual

Ever wanted to create a simple 2D platform game like the 80's classic Manic Miner? Or wanted to tweak or even create your own levels? You're in the right place, Welcome to the WonkyPix 2D Platform Gaming Engine. The engine is still under development and ©2005 Jason Brooks, this document describes the current features and limitations for the current version of the Engine.

We hope you have fun creating and modifying your own sprites and levels. Until the front end UI editor has been created, you will need a bit of technical knowledge to get the most out of this and hopefully this manual will guide you through the engines configuration.

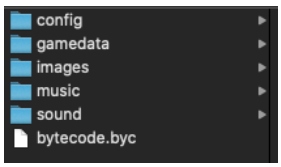
2 Introduction

The 2D Gaming Platform Engine is a simple configuration file driven way to create platform games similar in style to the 80's eight Bit Classic, it's easy to use, configure and with your imagination you could create some amazing old school gaming classics.

This document describes the tags used to configure the game and create your own levels, or even write your own Level editor if you're up to the task?

3 The Media Structure

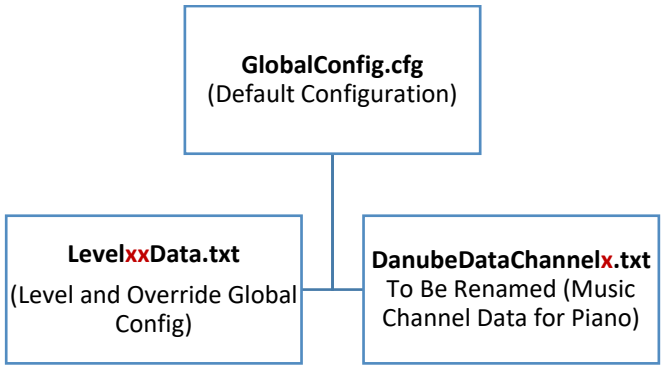
The media folders are all case sensitive and should be left intact, all folders contain configurable items.



3.1 Config

The config folder contains: -

- Global Configuration
- Level Data
- Channel data for the Piano loading screen
    - You can change the title page music and the piano keys which are played.
    - Supporting up to 10 channels of note playing.



4 GlobalConfig.cfg

The configuration file is mandatory and contains the games defaults (some of which can be overridden at the Level file). The file can be created and edited with your favourite plain text editor, like UltraEdit, BBEdit, Notepad etc.

An example can be found in Section 22

The configuration file is a text editable file and consist of

- Comments – Use a semicolon ;
- Tags – [TAG]
- Data – As described for each Tag

The Game Engine Parser is very basic and, in many instances, data related to a tag must be the very next line (no comments or blank lines allowed). It may be something I'll change in a future update, however if the game engine isn't behaving as you expect, then check the configuration first.

Many of the Tags are also case sensitive, again ensure spelling and accuracy.
For example, these are invalid

```
[NumberOfLives]
;2
3
```

```
[ NumberOfLives ]
3
```

```
[NUMBEROfLives]
3
```

This is Valid

```
[NumberOfLives]
5
```

4.1 [CollectibleSound]

Tag:	`[CollectibleSound]`
Data:	`String (Case Sensitive) – Filename of WAV Encoded Sound File`
Usage:	`[CollectibleSound]` `pick.wav`
Description:	Sets the Default sound made when our hero collects an object. Note that each level configuration file can override this default. This file must be located in the **./media/sound/** folder and is case sensitive. If the music doesn't play as expected, check the file is located in the **./media/sound/** Folder and is spelt correctly.

4.2 [DefaultCollectablePoints]

Tag:	`[DefaultCollectablePoints]`
Data:	`Integer`
Usage:	`[DefaultCollectablePoints]` `100`
Description:	Sets the Default points awarded when our Hero collects an object.

4.3 [DefaultCustomSpriteSheet]

Tag:	`[DefaultCustomSpriteSheet]`
Data:	`String (Case Sensitive) – Filename of the Custom Sprite Sheet to use.  Must be PNG File Format`
Usage:	`[DefaultCustomSpriteSheet]` `CustomSpriteSheet1.png`
Description:	Let's you add a custom sprite sheet for use within the game to add extra sprites and artwork. This file must be located in the **./media/images/** folder and is case sensitive. If the images aren't displayed as expected, check the file is located in the **./media/images/** Folder and is spelt correctly.

4.4 [DefaultDemoLevelMusic]

Tag:	`[DefaultDemoLevelMusic]`
Data:	`String (Case Sensitive) – Filename of OGG Encoded Music File`
Usage:	`[DefaultDemoLevelMusic]` `MountainKingOriginal.ogg`
Description:	Sets the Default music to use for all game levels when in Demo Mode. Note that each level configuration file can override the default. This file must be located in the ./media/music/ folder and is case sensitive. If the music doesn't play as expected, check the file is located in the ./media/music/ Folder and is spelt correctly.

4.5 [DefaultFinalCavernMusic]

Tag:	`[DefaultFinalCavernMusic]`
Data:	`String (Case Sensitive) – Filename of OGG Encoded Sound File`
Usage:	`[DefaultFinalCavernMusic]` `CavernComplete.ogg`
Description:	Sets the Default music to be played should our hero complete the final cavern (defined in the Level Configuration). Note that each Final level configuration file can override this default. This file must be located in the **./media/music/** folder and is case sensitive. If the music doesn't play as expected, check the file is located in the **./media/music/** Folder and is spelt correctly.

4.6 [DefaultHeroDeathHeight]

Tag:	`[DefaultHeroDeathHeight]`
Data:	`Integer`
Usage:	`[DefaultHeroDeathHeight]` `5`
Description:	Sets the number of blocks our hero can fall before losing a life.

4.7 [DefaultHighScoreMusic]

Tag:	`[DefaultHighScoreMusic]`
Data:	`String (Case Sensitive) – Filename of OGG Encoded Sound File`
Usage:	`[DefaultHighScoreMusic]` `EineKleineNachtMusik.ogg`
Description:	Sets the Default music to be played should our hero achieve a completed level highscore. Note that each level configuration file can override this default. This file must be located in the **./media/music/** folder and is case sensitive. If the music doesn't play as expected, check the file is located in the **./media/music/** Folder and is spelt correctly.

4.8 [DefaultLevelMusic]

Tag:	`[DefaultLevelMusic]`
Data:	`String (Case Sensitive) – Filename of OGG Encoded Music File`
Usage:	`[DefaultLevelMusic]` `InTheHallOfTheMountainKing.ogg`
Description:	Sets the Default music to use for all game levels. Note that each level can override the default and provide its own music for extra level depth. This file must be located in the ./media/music/ folder and is case sensitive. If the music doesn't play as expected, check the file is located in the ./media/music/ Folder and is spelt correctly.

4.9 [DefaultSwitchSoundOff] – Not Used (For Future Release)

Tag:	`[DefaultSwitchSoundOff]`
Data:	`String (Case Sensitive) – Filename of WAV Encoded Sound File`
Usage:	`[DefaultSwitchSoundOff]` `pick.wav`
Description:	Sets the Default sound made when our hero uses a switch to the off position. Note that each level configuration file can override this default. This file must be located in the **./media/sound/** folder and is case sensitive. If the music doesn't play as expected, check the file is located in the **./media/sound/** Folder and is spelt correctly.

4.10 [DefaultSwitchSoundOn]

Tag:	`[DefaultSwitchSoundOn]`
Data:	`String (Case Sensitive) – Filename of WAV Encoded Sound File`
Usage:	`[DefaultSwitchSoundOn]` `pick.wav`
Description:	Sets the Default sound made when our hero uses a switch. Note that each level configuration file can override this default. This file must be located in the **./media/sound/** folder and is case sensitive. If the music doesn't play as expected, check the file is located in the **./media/sound/** Folder and is spelt correctly.

4.11 [DisablePlaybackControl]

Tag:	`[DisablePlaybackControl]`
Data:	`Boolean (YES / NO) Defaults to NO if the TAG is missing.`
Usage:	`[DisablePlaybackControl]` `YES`
Description:	The game engine features the ability to show our Hero how to complete a cavern from pre-recorded game sessions, you can also (at the cost of a remaining life) have our hero complete a level you are stuck on and move to the next. You will need the Developer Licenced Engine to be able to create the `[ReplayData]` required for the replay functionality to work correctly.

4.12 [ExtraLifeSound]

Tag:	`[ExtraLifeSound]`
Data:	`String (Case Sensitive) – Filename of WAV Encoded Sound File`
Usage:	`[ExtraLifeSound]` `ExtraLife.wav`
Description:	Sets the Default sound to be played when our hero gains an extra life. Note that each level configuration file can override this default. This file must be located in the **./media/sound/** folder and is case sensitive. If the music doesn't play as expected, check the file is located in the **./media/sound/** Folder and is spelt correctly.

4.13 [FinalCollectibleSound]

Tag:	`[FinalCollectibleSound]`
Data:	`String (Case Sensitive) – Filename of WAV Encoded Sound File`
Usage:	`[FinalCollectibleSound]` `pick.wav`
Description:	Sets the Default sound made when our hero collects the final object on the level. Note that each level configuration file can override this default. This file must be located in the **./media/sound/** folder and is case sensitive. If the music doesn't play as expected, check the file is located in the **./media/sound/** Folder and is spelt correctly.

4.14 [NumberOfLevels]

Tag:	`[NumberOfLevels]`
Data:	`Numeric`
Usage:	`[NumberOfLevels]` `20`
Description:	Tell the engine the maximum number of levels your game has. Each Level Data file must be sequentially numbered and covered off later in this guide.

4.15 [NumberOfLives]

Tag:	`[NumberOfLives]`
Data:	`Numeric`

Usage:	`;`
	`; Set the initial start number of lives for the game`
	`;`
	`[NumberOfLives]`
	`3`

Description:	Set the start number of lives for the game. You could just have only one life, or 255.

4.16 [PianoMusic]

Tag:	`[PianoMusic]`
Data:	`String (Case Sensitive) – Filename of OGG Encoded Music File`
Usage:	`[PianoMusic]` `Danube.ogg`
Description:	Sets the Title Page music to use (on the Piano Screen). This points to a file in the **./media/music/** folder and is case sensitive. If the music doesn't play as expected, check the file is located in the ./media/music/ Folder and is spelt correctly.

4.17 [ProgramName]

Tag:	`[ProgramName]`
Data:	`String`
Usage:	`[ProgramName]` `ManicMiner`
Description:	Sets the program name string at the fake ZX Spectrum Loading Page. 

4.18 [SplashScreens]

Tag:	`[SplashScreens]`
Data:	`String (Case Sensitive) – Filename of the Custom Splash Screens to use. Must be PNG File Format`

Usage:	`[SplashScreens]`
	`CustomSplashScreen1.png`
	`Instructions.png`
	`AdditionalInfo.png`

Description:	Adds additional splash screens when the game engine loads up which allows additional info/credits to be displayed. This file must be located in the **./media/images/** folder and is case sensitive. If the images aren't displayed as expected, check the file is located in the **./media/images/** Folder and is spelt correctly.

4.19 [TargetExtraLifeScore]

Tag:	`[TargetExtraLifeScore]`
Data:	`Integer`
Usage:	`[TargetExtraLifeScore]` `10000`
Description:	Sets the Default player score target before the first extra life is awarded. If no extra lives are permitted use the value **0** (Number Zero)

4.20 [TargetExtraNextLifeScore]

Tag:	`[TargetExtraNextLifeScore]`
Data:	`Integer`
Usage:	`[TargetExtraNextLifeScore]` `15000`
Description:	Sets the Default additional score after the first extra life is awarded target before the first extra life is awarded. If no extra lives are permitted use the value **0** (Number Zero) For example :- If TargetExtraLifeScore = 10000 And TargetExtraNextLifeScore = 15000 Next lives will be awarded when our hero reaches 10000, 25000, 40000 …

5 Level Configuration Data

The following describes the tags and usage for each level data configuration file. Level files follow the following naming standard.

```
LevelxData.txt
```

Where x is numeric starting at 1, leading zeroes are not supported.

i.e. **Level01Data.txt** is invalid and will not be read by the level engine.

Level1Data.txt is valid and will attempted to be parsed by the level engine.

5.1 [Actors]

Tag:	`[Actors]`
Data:	`ImageName + Parameters`
Usage:	`[Actors]` `Phil.png, 8, 7, 1, 8, 7, 15, 7, 16, 0, 4, 4, 1, 8, 8, 8, 4`
Description:	See section 12 for more details on parameter usage. Multiple actors can be defined by adding more Actor information on each line directly after the TAG See section 23 for an example of level usage.

5.2 [BackgroundColour]

Tag:	`[BackgroundColour]`
Data:	`String (All Uppercase)`
Usage:	`[BackgroundColour]` `BLACKCOLOUR`
Description:	Sets the level background colour by name or CUSTOM RGB as defined in Colour Names Supported String Names Section 11

5.3 [BackGroundStaticImages] – Optional

Tag:	`[BackGroundStaticImages]`
Data:	`ImageName + Parameters`
Usage:	`[BackGroundStaticImages]` `SpookyWindow1.png`
Description:	Add extra levels of artwork to your game level, to give it a more polished and custom feel, this TAG will add images behind the game level board. If the image is to be located in a CustomSprite Sheet you will need to precede the Image name with **#** otherwise the engine will attempt to load the filename located in `./media/images/` **Parameters** X, Y, Opacity, [Initial Rotation Angle, Rotate By Angle] Example, `#SpookyWindow2.png, 470, 65, 196, 5.0` Will place the image SpookyWindow2.png located in the CustomSprite Sheet at X:470 and Y:65 with an Opacity of 196 and angle of 5 degrees. See Section **6**

5.4 [BlockNOP] – Optional

Tag:	`[BlockNOP]`
Data:	`ImageName`
Usage:	`[BlockNOP]` `MyCustomBlock.png`
Description:	If you want to use a different block other than blank background colour for your game, you can set a bespoke BlockNOP see **Section 6.1 Parameter 1 - ImageName.png - Mandatory** for the filename format. The Block will always be resized to 32 x 32 pixels.

5.5 [BorderColour]

Tag:	`[BorderColour]`
Data:	`String (All Uppercase)`
Usage:	`[BorderColour]` `DARKREDCOLOUR`
Description:	Sets the level Border colour by name or CUSTOM RGB as defined in Colour Names Supported String Names Section 11

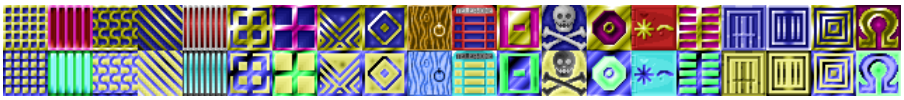
5.6 [CollectablePoints] – Optional

Tag:	`[CollectablePoints]`
Data:	`Integer`
Usage:	`[CollectablePoints]` `100`
Description:	Set the points value of collectibles for your current level (This will override the GlobalConfig Default).

5.7 [CustomSpriteSheet] – Optional

Tag:	`[CustomSpriteSheet]`
Data:	`Filename of PNG in media/images/ folder`
Usage:	`[CustomSpriteSheet]` `MyCustomSprites.png`
Description:	You can add additional sprites to your levels, by telling the engine where your custom sprite file and texture co-ordinates file is located.

5.8 [ExitBlock]

Tag:	`[ExitBlock]`
Data:	`Integer`
Usage:	`[ExitBlock]` `0`
Description:	There are 20 tile sets defined in the graphics block defined as 0 – 19 Set which block will be your level exit. 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 

5.9 [FinalCavernMusic] – Optional

Tag:	`[FinalCavernMusic]`
Data:	`String (Case Sensitive) – Filename of OGG Encoded Sound File`
Usage:	`[FinalCavernMusic]` `MyCustomSprites.png`
Description:	Play a custom OGG file when our hero completes a level marked by [SwordFish], this will override the [DefaultFinalCavernMusic This file must be located in the **./media/music/** folder and is case sensitive. If the music doesn't play as expected, check the file is located in **./media/music/** Folder and is spelt correctly.

5.10 [ForeGroundStaticImages] – Optional

Tag:	`[ForeGroundStaticImages]`
Data:	`ImageName + Parameters`
Usage:	`[ForeGroundStaticImages]` `SpookyWindow1.png`
Description:	Add extra levels of artwork to your game level, to give it a more polished and custom feel, this TAG will add images in front of the game level board. If the image is to be located in a CustomSprite Sheet you will need to precede the Image name with `#` otherwise the engine will attempt to load the filename located in `./media/images/` **Parameters** X, Y, Opacity, [Initial Rotation Angle, Rotate By Angle] Example, `#SpookyWindow2.png, 470, 65, 196, 5.0` Will place the image SpookyWindow2.png located in the CustomSprite Sheet at X:470 and Y:65 with an Opacity of 196 and angle of 5 degrees. See Section 6

5.11 [HeroDeathHeight] – Optional

Tag:	`[HeroDeathHeight]`
Data:	`Integer`
Usage:	`[HeroDeathHeight]` `5`
Description:	Set the maximum height in game blocks our hero can fall before losing a life.

5.12 [HeroStart]

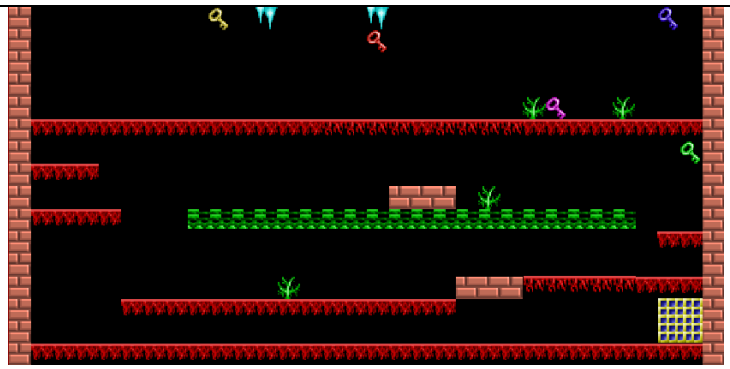
Tag:	`[HeroStart]`
Data:	`ImageName + Parameters`
Usage:	`[HeroStart]` `MinerWilly.png, 2, 13, 1, 16, 2, 4, 16`
Description:	See section **13** for more details on usage. Support for multiple Hero's has been added though a practical usage has yet to be found.

5.13 [HighScoreMusic] - Optional

Tag:	`[HighScoreMusic]`
Data:	`String (Case Sensitive) – Filename of OGG Encoded Sound File`
Usage:	`[HighScoreMusic]` `YourLevelMusic.ogg`
Description:	Plays a custom OGG file when our hero achieves a high score for the level, this will override the [DefaultHighScoreMusic], giving additional depth and atmosphere to your game. This file must be located in the **./media/music/** folder and is case sensitive. If the music doesn't play as expected, check the file is located in the **./media/music/** Folder and is spelt correctly.

5.14 [LevelData]

Tag:	**[LevelData]**
Data:	**GridArray**
Usage:	**[LevelData]** **30600000060000003003000000000003** **30000000000000000300300000000003** **300000000000090EE300000000000113** **30000000000000000300000000000003** **30000000000000000300000000000003** **31110000011111100311000000000003** **30000000000009003000111100000013** **30111000000000000300000000010003** **30900000111000000300000000000003** **30000000000000000311111000000003** **31000000000011100300000000011113** **30000000011000000D00000000000003** **30001100000000000D00001111100003** **30000000000CCC3AB311000000000903** **300000000000003BB300000500000003** **31111111111111111111111111111113**
Description:	The level data, currently only supports a fixed basic 32 blocks width x 16 blocks height and used to define the main static level layout. Error Checking is minimal and comments are not supported in the GridArray Block, nor are blank lines. It's just fussy at the moment… 0 = Empty Space 1 = Solid Platform Type 1 2 = Solid Platform Type 2 3 = Impassable Wall 4 = Collapsible Platform 5 = Hazard 1 6 = Hazard 2 7 = Conveyor Platform Moving Right To Left 8 = DO NOT USE 9 = Collectable Object A = Willy's Exit/Door Position (Top Left Corner) B = Special Control to detect Willy in the door C = Conveyor Platform Moving Left To Right D = Removable WALL Block E = Removable Platform Block 1 F = Special Block (Willy's Home Graphic) G = Removeable Floor Type 2 (RFU) **The top left corner denotes co-ordinates (0,0)**

5.15 [LevelMusic] - Optional

Tag:	`[LevelMusic]`
Data:	`String (Case Sensitive) – Filename of OGG Encoded Sound File`
Usage:	`[LevelMusic]` `YourLevelMusic.ogg`
Description:	Plays a custom music file (must be OGG File Encoded) for your level giving players additional depth and atmosphere to your game, if the tag is absent the default level music defined in GlobalConfig.cfg will be used. This file must be located in the **./media/music/** folder and is case sensitive. If the music doesn't play as expected, check the file is located in the **./media/music/** Folder and is spelt correctly.

5.16 [LevelName]

Tag:	`[LevelName]`
Data:	`String`
Usage:	`[LevelName]` `The Central Cavern`
Description:	Define the name of the level to be shown on the game bar. 

5.17 [LevelOnDeath] – Optional

Tag:	`[LevelOnDeath]`
Data:	`Integer`
Usage:	`[LevelOnDeath]`
Description:	The bastard parameter, should a player not complete the level you have the choice of sending the player to a different level in the game, or even backwards if the level design contains a series of challenges that must be completed in order. You could create a personal Hell level that must be completed on death… Use the level number 1 – Maximum Number of levels you have defined in your game.

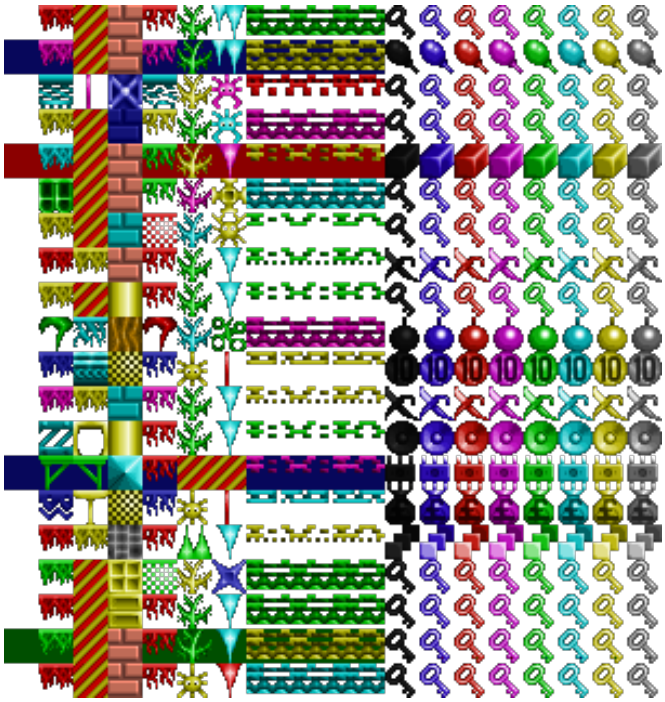
5.18 [ReplayData] – Optional

Tag:	`[ReplayData]`
Data:	`String`
Usage:	`[ReplayData]`
Description:	Only available to developers that have licenced the game engine, this contains a string of data that enables our hero to demonstrate how the game level can be completed without losing a life.

5.19 [SpecialFunction] - Optional

Tag:	`[SpecialFunction]`
Data:	`String + Parameters`
Usage:	`[SpecialFunction]` `SolarPoweredGenerator, 24 ,1, 210, WHITECOLOUR`
Description:	Add special functions to your level design. See Section 16 Special Functions for more information.

5.20 [SpriteSet]

Tag:	`[SpriteSet]`
Data:	`Integer`
Usage:	`[SpriteSet]` `0`
Description:	There are 20 tile sets defined in the graphics block defined as 0 – 19 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 

5.21 [StickyConveyor] – Optional

Tag:	`[StickyConveyor]`
Data:	`Integer`
Usage:	`[StickyConveyor]` `2`
Description:	This value determines the type of conveyor belt platform our hero lands on. Valid Values are 0 – Normal Conveyor Action (Once our hero stops, movement will change to the conveyor direction). 1 – Rigid Action, will force direction when our hero lands. 2 – Height Rule, movement will depend on height of our hero when landing on the conveyor. Feel free to experiment with this setting to fine-tune your levels.

5.22 [Switches] – Optional

Tag:	`[Switches]`
Data:	`ImageName.png + Parameters as CSV`

Usage:	`[Switches]`
	`Switches.png,  6, 0, 0, Switch1, OpenTrapDoor, 18:12:18:13 , pick.wav`
	`Switches.png, 18, 0, 0, Switch2, OpenTrapDoor, 16:3:17:3`

Description:	This Tag adds named switches to your game level that can be used to trigger various functions. Basic Parameters are 1. Actor File Name 2. Block X 3. Block Y 4. Start State 5. Special Function 6. Special Function Parameters 7. Optional Switch Sound See Section 17 for more information.

5.23 [SwitchSoundOff] – Not Yet Implemented - RFU

Tag:	`[SwitchSoundOff]`
Data:	`String (Case Sensitive) – Filename of WAV Encoded Sound File`
Usage:	`[SwitchSoundOff]` `LevelCustomSwitchSoundOff.wav`
Description:	Changes the default sound (Only for the level) of switches when they are clicked on. A custom sound file (must be WAV File Encoded) for your level giving players additional depth and atmosphere to your game, if the tag is absent the default switch sound defined in GlobalConfig.cfg will be used. This file must be located in the **./media/sound/** folder and is case sensitive. If the music doesn't play as expected, check the file is located in **./media/sound/** Folder and spelled correctly.

5.24 [SwitchSoundOn] – Optional

Tag:	`[SwitchSoundOn]`
Data:	`String (Case Sensitive) – Filename of WAV Encoded Sound File`
Usage:	`[SwitchSoundOn]` `LevelCustomSwitchSoundOn.wav`
Description:	Changes the default sound (Only for the level) of switches when they are clicked on. A custom sound file (must be WAV File Encoded) for your level giving players additional depth and atmosphere to your game, if the tag is absent the default switch sound defined in GlobalConfig.cfg will be used. This file must be located in the **./media/sound/** folder and is case sensitive. If the music doesn't play as expected, check the file is located in **./media/sound/** Folder and spelled correctly.

5.25 [SwordFish] – Optional

Tag:	`[SwordFish]`
Data:	`Image Name + Parameters`
Usage:	`[SwordFish]` `SwordFish.png, 19, 5, 19, 2, 4`
Description:	The presence of this Tag indicates a final cavern and our hero is free.

Parameter	Value	Description
Image Name	SwordFish.png	Point to an image that will be used to show the final end of level. This sprite currently must reside in the Main Sprite Sheet and not in a custom sprite sheet.
SwordFishX	19	Using Block co-ordinates, positions the Top Left X-Position of the Swordfish Block Image. (Usually the exit block)
SwordFishY	5	Using Block co-ordinates, positions the Top Left Y-Position of the Swordfish Block Image. (Usually the exit block)
HeroX	19	Using Block co-ordinates, positions our Hero at the Top Left X-Position on the level map.
HeroY	2	Using Block co-ordinates, positions our Hero at the Top Left Y-Position on the level map.
Hero Frame Number	4	Which Image in the Animation Sequence should our Hero display.

5.26 [SyncFlash] – Optional

Tag:	`[SyncFlash]`
Data:	`None`
Usage:	`[SyncFlash]`
Description:	No parameters for this tag, if present in the level file, all collectibles will flash the same colour at the same time, instead of cycling colours.

5.27 [WillyDieSound] – Optional

Tag:	`[WillyDieSound]`
Data:	`String (Case Sensitive) – Filename of WAV Encoded Sound File`
Usage:	`[WillyDieSound]` `YourCustomDieSound.wav`
Description:	Plays a custom WAV file when our hero loses a life, giving additional depth and atmosphere to your game, if the tag is absent the default Die Sound defined in GlobalConfig.cfg will be used. This file must be located in the **./media/sound/** folder and is case sensitive. If the music doesn't play as expected, check the file is located in the **./media/sound/** Folder and is spelt correctly.

5.28 [ZHero] – Optional

Tag:	`[ZHero]`
Data:	`No Data`
Usage:	`[ZHero]`
Description:	The presence of this Tag ensures our Hero moves behind Switches, by default our Hero moves in front of Switches in the Z-Order of the level.

5.29 [ZActor] – Optional

Tag:	`[ZActor]`
Data:	`No Data`
Usage:	`[ZActor]`
Description:	The Presence of this Tag ensures that Actors are drawn in-front of switches instead of behind switches as default behaviour.

6 Static Image Parameter Format

The following section describes the static image parameter format used for both Foreground and Background images applied to the game scene.

The parameter consists of four Mandatory Parameters and two Optional Parameters.

The command format is :-

```
 ImageName.png, X, Y, Opacity, [ Initial Rotation, Rotate By ]
#ImageName.png, X, Y, Opacity, [ Initial Rotation, Rotate By ]
```

Example:

`Window1.png, 0, 0, 255`	Place Image Window1.png at **(0, 0)** [Top Left Corner] fully Opaque
`Window1.png, C, 0, 128, 25.0`	Place Image Window1.png at **(Centre of Screen Horizontal, 0)** Semi Transparent with a rotation of 25 Degrees
`Window1.png, C, C, 255, -25.0, 2.2`	Place Image Window1.png at (Centre of Screen Horizontal, Centre of Screen Vertically) Semi Transparent with an initial rotation of -25 Degrees (or 335 Degrees) and apply a continuous rotation of 2.2 Degrees per frame.

6.1 Parameter 1 - ImageName.png - Mandatory

This can be an Image in a Texture Atlas or a physical file located in ./media/images/

To identify an Image in a Texture Atlas use **#** at the start of the filename.

6.1.1 Usage with a Custom Sprite Sheet

Example:

```
#SpookyWindow.png
```

Will point to a texture called SpookyWindow.png within the CustomSprite Sheet Defined either at the GlobalConfig.cfg or Local Level File.

It is not the same as SpookyWindow.png which will look for a physical file.

6.1.2 Usage with a file image

Example:

```
SpookyWindow.png
```

Will load a texture called SpookyWindow.png from ./media/images/
This is case sensitive.

It is not the same as #SpookyWindow.png which will look for a Texture within an Atlas.

6.2 Parameter 2 – X Position (Integer or C) - Mandatory

The Game Board 0,0 co-ordinate is Top Left of the Screen and refers to the number of Pixels (0-639) to place your Image on the horizontal plane.

If you wish to centre your image on the X (Horizontal Plane), you can also use the value **C**

6.3 Parameter 3 – Y Position (Integer or C) - Mandatory

The Game Board 0,0 co-ordinate is Top Left of the Screen and refers to the number of Pixels (0-639) to place your Image on the horizontal plane. Only Integers supported

6.4 Parameter 4 – Opacity (Integer 0-255) - Mandatory

Set the opacity of your image on screen, range is 0 – 255 where 0 is fully translucent and 255 is fully opaque.

6.5 Parameter 5 – Initial Rotation (Float) - Optional

Set the rotation in Degrees for the texture you wish to add.

6.6 Parameter 6 – Rotate By (Float) - Optional

You can set the image to automatically rotate by an offset amount for each frame rendered. +ve for clockwise, -ve for counter clockwise.

You could create an image like a psychedelic swirl that spins endlessly in the background for example.

Note Parameter 5 must be present to use this feature.

7 Block Data Types and their Usage

Block Data for the platform game grid have different uses.

Block Type ID	Type	Description
0	Blank Space	As the type suggests, doesn't render any data to the screen[1] and will leave the background as per the background colour set.
1	Solid Platform Block 1	This selects the first Platform block from the list. You can pass through (Upwards of a Platform Block) but our Hero's feet when landing on this block type will stop falling.
2	Solid Platform Block 2	This selects the Second Platform block from the list (you can use this to vary the style a little). You can pass through (Upwards of a Platform Block) but our Hero's feet when landing on this block type will stop falling.
3	Impassable Wall	Think of this like a brick wall, you can neither jump through it, nor pass through it and is an effective barrier for your level.
4	Collapsible Platform	This platform block type will collapse when our hero walks or makes contact with the block type, once the platform disappears there the game map will be replaced with a Type 0 game block.
5	Hazard 1	This block creates a hazard that will make our hero lose a life and restart the level[2].
6	Hazard 2	This block creates a hazard that will make our hero lose a life and restart the level and works the same as Type 5.
7	Conveyor Platform Moving Right to Left	Creates a moving platform block that fill force willy to move in one direction.
8	Reserved for internal use DO NOT USE	This block is reserved and for internal use only.
9		
A	Top Left of Exit Block Marker	This block type must be the Top Left of any exit indicator block. Exit Block Indicators are denoted by the block **AB** **BB**
B	Trigger Exit Marker	This block type must be Top Right, Bottom Left or Bottom Right and is used to check if our Hero has triggered the exit. For example, the block **AB** 00

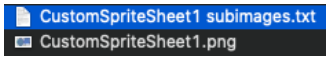
[1] Except where BlockNOP Parameter is used on a game level.
[2] Level will restart at the beginning unless the [LevelOnDeath] tag is used.

		Will trigger the end of level (once all collectibles are retrieved) when our hero enters the exit and collision detection detects the hero is inside the box, or dropped into the box from above.
		
		The block type
		
		`A0`
		`BB`
		
		Can be used when our hero is fully in the exit (when falling from above).
		
		This is used to good effect in the level "Processing Plant" which means our hero must touch the top right corner of the exit on a jump making it a difficult exit.

C	Conveyor Platform moving Left to Right	Creates a moving platform block that fill force willy to move in one direction.
D	Removeable Wall	A special block type that can be removed with the use of a Switch
E	Removeable Floor	A special block type that can be removed with the use of a Switch
F	Special Block	For each block encountered on the Grid it will take a copy of the Home Page and place the block in situ. This block type acts like a wall in that our hero cannot pass through this block type. Code needs reviewing, don't think this is working as expected.
G	Removeable Floor Type 2	Reserved for future use, currently not correctly implemented as intended.

8 Custom Sprite Files Format

The game engine will allow you add additional sprite sheets in addition to the default ones. You will need to create a PNG of all the textures you plan to use and a subimages file with texture co-ordinates.



It is important that the subimages file is named identical to the main image file with (space subimages.txt) as show in the example above.

The subimages.txt file must contain the name of the Image and co-ordinates in the following format.

```
TextureName.png:X:Y:WIDTH:HEIGHT
```

```
BlockNOP.png:977:645:32:32
SpinnerBackGround1.png:851:3:640:640
Swirl.png:3:3:846:846
Window1.png:851:645:61:173
Window2.png:914:645:61:173
```

Using a software package like https://www.codeandweb.com/texturepacker will make creating custom sprites sheets simple and easy for your projects.

9 Colour String Names – Backgrounds

In addition to the default colour names provided below, you can use a string for the RGB Values to customise the look.

```
CUSTOM:RED:GREEN:BLUE
```

Where Red, Green, Blue values range from 0 – 255

For example :-

```
CUSTOM:255:0:255
```

Will result in a Purple Colour

See Colour Names Supported String Names Section 11

10 Colour String Names – Sprites

In addition to the default colour names provided below, you can use a custom string for the RGB Values to customise the look of a sprite, the table is similar to backgrounds with the addition of the TRANSPARENT colour which sets the sprite to invisible.

```
CUSTOM:RED:GREEN:BLUE
```

Where Red, Green, Blue values range from 0 – 255

For example :-

```
CUSTOM:255:0:255
```

Will result in a Purple Colour.

See Colour Names Supported String Names Section 11

11 Colour Names Supported String Names

Colour String Name	R	G	B
BLACKCOLOUR	0	0	0
BLUECOLOUR	0	0	255
CYANCOLOUR	0	255	255
DARKBLUECOLOUR	7	7	91
DARKCYANCOLOUR	0	191	191
DARKGREENCOLOUR	0	79	0
DARKMAGENTACOLOUR	161	47	190
DARKREDCOLOUR	139	0	0
DARKYELLOWCOLOUR	188	190	0
GREENCOLOUR	0	255	0
GREYCOLOUR	128	128	128
LIGHTGREYCOLOUR	192	192	192
MAGENTACOLOUR	255	0	255
REDCOLOUR	255	0	0
VERYDARKBLUECOLOUR	3	3	64
VERYDARKCYANCOLOUR	0	121	121
VERYDARKGREENCOLOUR	0	64	0
VERYDARKMAGENTACOLOUR	121	23	121
VERYDARKREDCOLOUR	65	0	0
VERYDARKYELLOWCOLOUR	121	121	0
WHITECOLOUR	255	255	255
YELLOWCOLOUR	255	255	0
CUSTOM	**0-255**	**0-255**	**0-255**
TRANSPARENT (ALPHA = 0) Supported on Sprites only	0	0	0

12 Actor Parameter Format

The following describes the Actor Parameter Format, care must be taken on the type of animation graphics used, but assumes the fixed block type provided with the engine.

12.1 Example Usage

```
[Actors]
Phil.png, 8, 7, 1, 8, 7, 15, 7, 16, 0, 4, 4, 0, 8, 8, 8, 4, Optional
```

Parameter	Value	Description
Actor Name	Phil.png	Point to an animation image Sequence there are Eight frames of animation here. 
Start Block X Position	8	Top Left Block X-Coordinate of our Actor
Start Block Y Position	7	Top Left Block Y- Coordinate of our Actor
Initial Direction	1	The Frame Number the actor will start with when the game is initialised.
Minimum Block X Position	8	The Minimum X range our actor can move
Minimum Block Y Position	7	The Minimum Y range our actor can move
Maximum Block X Position	15	The Maximum X range our actor can move
Maximum Block Y Position	7	The Maximum Y range our actor can move
Delta X in Pixels per movement	16	The number of pixels our actor will move when the animation tick count is exhausted. Since our actor is defined on 32 pixels per frame (Simulating old school systems) we'll move 16 pixels to keep the animation fluid.
Delta Y in Pixels per movement	0	The number of pixels our actor will move when the animation tick count is exhausted.
Tick Count	4	The number of render frames to wait before updating the animation.
Left Frame	4	State where the Left Frame Animation of the Actor starts, in Phil's case it is Frame 4
Right Frame	0	State where the Right Frame Animation starts for our actor.
Up Frame	8	State where the Upmost Frame Animation of the Actor starts (unused [no **dy** movement] set to 8)
Down Frame	8	State where the Bottommost Frame animation of the Actor start (unused [no **dx** movement] set to 8)
maxFrames	8	The maximum number of frames of animation in the image black to be used.
Frames Per Animation Segment	4	How fast the actor should be updated, this defines the number of frames to skip before updating occurs.
Action (Optional Parameter)	String	A valid string of an in game routine to apply additional control to the actor.
Linked to Switch (Optional Parameter)	String	A valid string matching the name of a [SWITCH] that has been configured.

13 Optional Actor In-Game Action Routines

The following describes additional optional parameters that can be used to control in-game actors.

Parameter	Action
EUGENE	When all collectibles are collected, it forces the Actor to change to the downward direction and move to the Maximum Y block position.
SATELLITE	Used for Actors moving on the Y-Axis. When the Actor reaches the Maximum Y Position it will play the sprite animation and then restart at the minimum Y Position and new X position based on an internal algorithm.
KONG1	When activated by a switch, will drop the actor to the ground (Maximum Y)
ADDKONGPOINTS	Used in Conjunction with **DropTheBeast** will add the default collectible score for each block the actor falls.

14 Optional Actor Functions

The following describes additional parameters that Switches can be used to dynamically change the game level.

14.1 UpdateBlockParms

This function is used to update the Block Range of an Actors movement when a matching named switch has been triggered. For example, when a switch is activated, and actor could be set to start moving, or change the range of movement if a section of floor or wall is removed.

Actor Special Function	Parameter	Action
UpdateBlockParms	Min Block X	The new Minimum Block X Position the Actor can Move.
	Min Block Y	The new Minimum Block Y Position the Actor can Move.
	Max Block X	The new Maximum Block X Position the Actor can Move.
	Max Block Y	The new Maximum Block Y Position the Actor can Move.

Usage Example (Taken from Level8Data.txt) : -

```
MagentaBarrel.png, 11, 11, 1, 11, 11, 15, 11, 16, 0, 8, 4, 1, 8, 8, 4, 4, ,
Switch1, UpdateBlockParms, 11:11:18:11
```

Changes the Actor from moving between Block Co-Ordinates 11 -> 15 on the X-Axis to 11 -> 18 on the X-Axis once the removeable wall was triggered.

14.2 DoKongDrop

This special routine will drop an actor to the game block co-ordinates you define and change the Actor Image over, once the Actor reaches the maximum Y position, it will be removed from the Game Board.

Actor Special Function	Parameter	Action
DoKongDrop	ImageName.png	Change the Actor Image to a new Image from the Spriteset (Currently must be in the main spritesheet)
	StartBlockX	Set the Game Grid X Position of the updated Actor
	StartBlockY	Set the Game Grid Y Position of the updated Actor
	MaxBlockX	Set the Maximum Game Grid X Position of the updated Actor
	MaxBlockY	Set the Maximum Game Grid Y Position of the updated Actor
	ActorDX	Set the X offset by which the Actor moves
	ActorDY	Set the Y offset by which the Actor moves
	TotalFrames	Number of Frames in the animation
	FramePerSegment	How many frames to wait between animation
	SpecialFunction	Optional Actor Special Function

Usage example (Taken from Level8Data.txt): -

```
GreenKong.png,      15,  0, 1, 15,  0, 15,  0,   0, 0, 45, 4, 1, 8, 8, 2, 2, ,
Switch2, DoKongDrop, MagentaKong.png:15:0:15:13:0:8:2:4:AddKongPoints
```

In this example, the GreenKong Image  is replaced with the MagentaKong  set of images (two animation frames), that will remain at X Position 15 and will move from Y position 0 to 13 moving 8 pixels every four frames.

14.3 AddKongPoints

This function will add points to the player score for each movement. Typically used in conjunction with DoKongDrop as points are limited to how long the actor remains on screen.

Usage example (Taken from Level8Data.txt): -

```
GreenKong.png,      15,  0, 1, 15,  0, 15,  0,   0, 0, 45, 4, 1, 8, 8, 2, 2, ,
Switch2, DoKongDrop, MagentaKong.png:15:0:15:13:0:8:2:4:AddKongPoints
```

Note, DoKongDrop allows an additional Special Function and designed for the AddKongPoints. Other special functions may result in unpredictable behaviour.

15 Hero Parameter Format

The following describes the Hero Actor Parameter Format, care must be taken on the type of animation graphics used, but assumes the fixed block type provided with the engine.

Example Usage

```
[HeroStart]
MinerWilly.png, 2, 13, 1, 16, 2, 4, 16
```

Parameter	Value	Description
Actor Name	MinerWilly.png	Point to an animation image Sequence 
Start Block X Position	2	Top Left Block X-Coordinate of our Actor
Start Block Y Position	13	Top Left Block Y- Coordinate of our Actor
Initial Direction	1	The Frame Number the actor will start with when the game is initialised.
Delta X in Pixels per movement	16	The number of pixels our actor will move when the animation tick count is exhausted. Since our actor is defined on 32 pixels per frame (Simulating old school systems) we'll move 16 pixels to keep the animation fluid.
Delta Y in Pixels per movement	2	The number of pixels our actor will move when the animation tick count is exhausted.
Tick Count	4	The number of render frames to wait before updating the animation.
Frames Per Animation Segment	16	The maximum number of animation frames for our hero.

16 Special Functions

The following special functions can be added to a level.

16.1 SolarPoweredGenerator

The Solar Powered Generator will draw a block of light starting from X, Y on the game board and beam directly down to the bottom of the screen. If the beam of light hits an Actor, it will deflect to the left, or down and will stop when the beam hits the bottom of the screen or a platform block.

If the block of light passes through our Hero, the level air will be lost at a fast rate than normal.

Only One Solar Powered Generator maybe active per level.

Usage:

```
SolarPoweredGenerator, 24 ,1, 210, WHITECOLOUR
```

Parameter	Value	Description
Start Block X Position	24	Top Left Block X-Coordinate of our solar powered generator.
Start Block Y Position	1	Top Left Block Y-Coordinate of our solar powered generator.
Opacity	210	Valid Values 0 – 255, where 0 is transparent and 255 is fully opaque.
Block Colour	WHITECOLOUR	The Colour of the Solar Powered Blocks

17 Switch Parameter Format and Functionality

Switch functionality when combined with Actors provide you additional flexibility and additional challenges that you can give your players.

Switches are flexible and simple to implement; however, your game script can become complex quickly if you don't approach them in a logical way.

17.1 Switch Tag Parameters

Parameter	Example	Description
Image Name	Switches.png	The name of two state PNG representing the type of switch/lever you want to use. First Image must be Off 
Block X	6	The X Position on the Game Grid the Switch is to be located.
Block Y	0	The Y Position on the Game Grid the Switch is to be located. This example has the Top of the Gameboard.
Switch Name	Switch1	The name of the switch, it can be any ASCII String (No Spaces) that can be used to match a switch with an Actor (if used).
Special Function	OpenTrapDoor	The name of the special function to trigger when the switch is activated. **Optional**
Special Function Parameters	18:12:18:13	A list of GameBlock Co-Ordinates to remove removeable GameBlocks from the game, in X:Y format. You can add multiple coordinates by simply stringing X:Y blocks together. i.e. `18:12` Block at (18, 12) `18:12:18:13` Blocks at (18,12) and (18,13) `18:12:18:13:6:2` Blocks at (18,12), (18:13) and (6,2) **Optional**
Switch Sound	Click.wav	A WAV file located in ./media/sound to be played when the switch is activated. Note this parameter will override the local and global Switch sound commands, giving you the ability to create more atmospheric sounds to your level design.

17.2 Switch Tag Examples

The following provides example Switch Usage Commands. Please note that you can add multiple switches to your level design, these must be added directly under the [Switches] Tag.

17.2.1 Simple Named Switch

The simplest form is to create a Switch which is named only. This example will usually be linked to an Actor which is "Listening" for the switch to be activated. The sound played will be the Gobal Config default unless overridden by the local level configuration.

```
Switches.png, 6, 0, 0, Switch1
```

17.2.2 Named Switch with Special Function

A switch can be linked to a special function and associated parameters

```
Switches.png, 6, 0, 0, Switch1, OpenTrapDoor, 18:12:18:13
```

17.2.3 Named Switch with Special Function and Different Sound

Each switch can be assigned a unique sound when activated using WAV files located in ./media/sound These are case sensitive, if the correct sound doesn't trigger when the switch is activated check the file exists and the filename matches.

```
Switches.png, 6, 0, 0, Switch1, OpenTrapDoor, 18:12:18:13, Click.wav
```

17.2.4 Named Switch No Special Function and Different Sound

If you want a switch to make a unique sound but not attached to a special function, you must use additional comma separators for the missing parameters.

Each switch can be assigned a unique sound when activated using WAV files located in ./media/sound These are case sensitive, if the correct sound doesn't trigger when the switch is activated check the file exists and the filename matches.

```
Switches.png, 6, 0, 0, Switch1, , , Click.wav
```

17.2.5 Adding Multiple Switches to the Level

You will usually want more than one switch active on the Gameboard at one time, and this can be achieved by adding multiple switch statements. Comments using the "`;`" are supported if you are trying new designs.

```
[Switches]
 Switches.png, 6,  0, 0, Switch1, OpenTrapDoor, 18:12:18:13 , Pull.wav
;Switches.png, 6,  0, 0, Switch1, , , Click.wav
 Switches.png, 18, 0, 0, Switch2, OpenTrapDoor, 16:3:17:3
```

18 Switch Special Functions

The following defines the currently implemented Special Functions for Switches.

Special Function	Parameters	Description
OPENTRAPDOOR	X:Y List	Removes a series of Game Blocks from the board effectively opening up a doorway for our hero to pass.
CLOSETRAPDOOR	X:Y List	Adds a series of Game Blocks at the coordinates in the parameter list. Can be used to add additional blocks to help our hero out of a level, or if triggered, block our hero from completing the level.

The X:Y parameter list may contain multiple X:Y Game Board Co-Ordinates, and must be separated by an extra colon `:`

For example

```
10:12
10:12:11:12
10:12:11:12:10:13:11:13
```

Provide various gameboard coordinates to the two Special Functions.

19 Actor Linked to Switch Special Functions

This feature can take some getting your head around from a parameters and usage point of view. Take the following use case scenario.

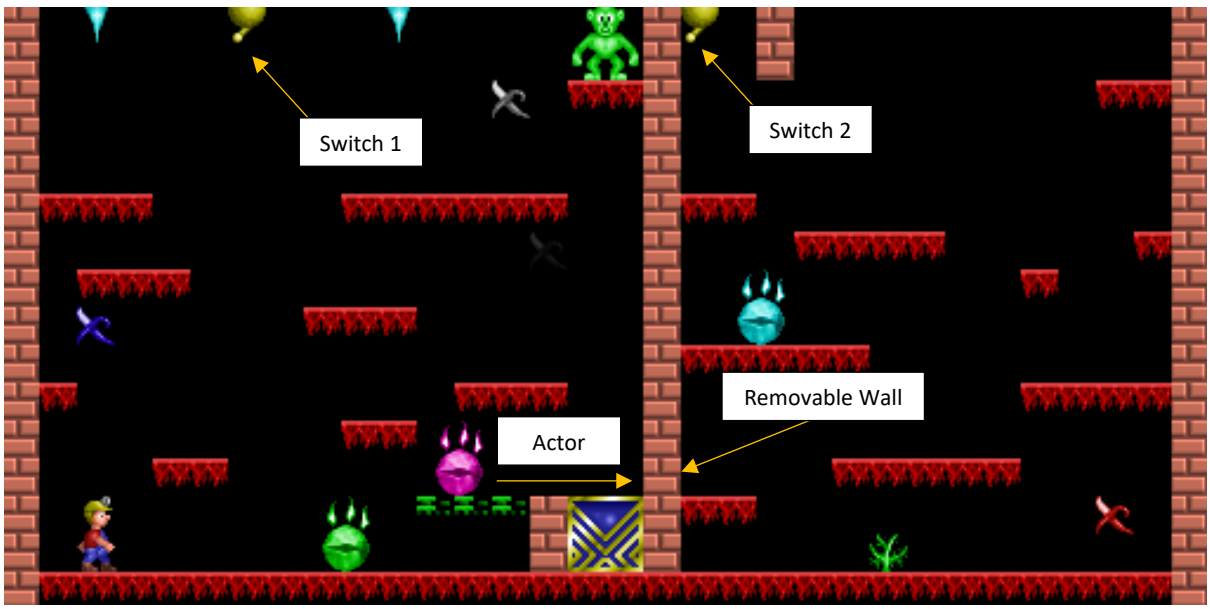
```
[Switches]
Switches.png,6,  0, 0, Switch1, OpenTrapDoor, 18:12:18:13 , Click.wav
Switches.png,18, 0, 0, Switch2, OpenTrapDoor, 16:3:17:3


[Actors]
GreenBarrel.png,     9,  13, 4,  1, 13,  9, 13, -16, 0,  4, 4, 1, 8, 8, 4, 4
BlueBarrel.png,      18,  7, 2, 18,  7, 21,  7,  16, 0,  4, 4, 1, 8, 8, 4, 4
MagentaBarrel.png,  11, 11, 1, 11, 11, 15, 11,  16, 0,  8, 4, 1, 8, 8, 4, 4, ,
Switch1, UpdateBlockParms, 11:11:18:11
GreenKong.png,       15,  0, 1, 15,  0, 15,  0,   0, 0, 45, 4, 1, 8, 8, 2, 2, ,
Switch2, DoKongDrop, MagentaKong.png:15:0:15:13:0:8:2:4:AddKongPoints
```

Here we have **Switch1** at (6,0) on the game board, when the switch is operated, the following takes place.

1. Open up the wall near the Magenta Barrel Actor
2. Update the parameters of the Magenta Barrel Actor so it knows it can move further along on the X-Axis.
3. Play the sound Click.wav from ./media/sound

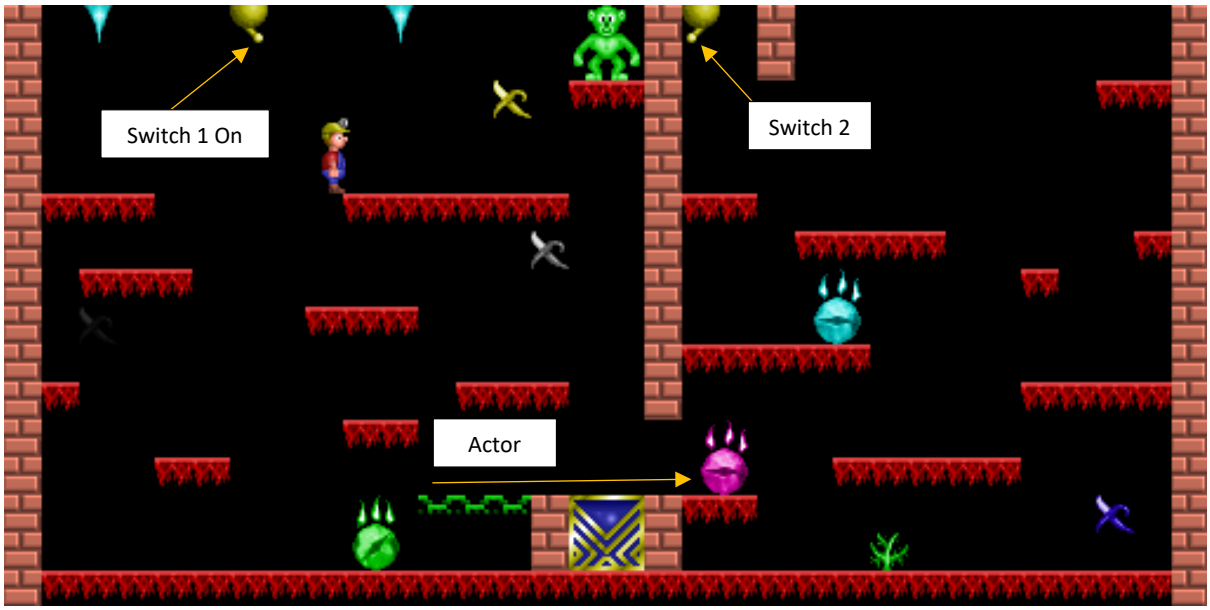
Using the configuration above in the orange highlighted commands, we have named our Switch `"Switch1"`

And linked our Actor "`MagentaBarrel.png`" to listen for "`Switch1`" to perform the function "`UpdateBlockParms, MinX:MinY:MaxX:MaxY`"



Once the switch has been activated, we can see that the removable wall "**Block Type D**" has been removed and the Actor has been updated to ensure it can move further across the game board.

Please note, that multiple Actors can Listen for the same Switch and be updated accordingly.



The next section, we want the floor below Kong to disappear when Switch 2 is activated, and for the Kong Beast to fall to the floor.

Switch 2 is configured to remove Game Blocks from the level to "enable" the Kong Beast to drop.

```
[Switches]
Switches.png, 6,  0, 0, Switch1, OpenTrapDoor, 18:12:18:13 , pick.wav
Switches.png, 18, 0, 0, Switch2, OpenTrapDoor, 16:3:17:3


[Actors]
GreenBarrel.png,     9,  13, 4,  1, 13,  9, 13, -16, 0,  4, 4, 1, 8, 8, 4, 4
BlueBarrel.png,      18,  7, 2, 18,  7, 21,  7,  16, 0,  4, 4, 1, 8, 8, 4, 4
MagentaBarrel.png,  11, 11, 1, 11, 11, 15, 11,  16, 0,  8, 4, 1, 8, 8, 4, 4, ,
Switch1, UpdateBlockParms, 11:11:18:11
GreenKong.png,       15,  0, 1, 15,  0, 15,  0,   0, 0, 45, 4, 1, 8, 8, 2, 2, ,
Switch2, DoKongDrop, MagentaKong.png:15:0:15:13:0:8:2:4:AddKongPoints
```

Our Actor GreenKong.png is listening for Switch2 to activate before executing the **DoKongDrop** function.

Here we have **Switch2** at (18,0) on the game board, when the switch is operated, the following takes place.

1. Remove the floor beneath the GreenKong Actor
2. Request the Special Function "`DoKongDrop`"
3. Set the DoKongDrop Parameters

Using the configuration above in the orange highlighted commands, we have named our Switch `"Switch1"`

And linked our Actor "`MagentaBarrel.png`" to listen for "`Switch1`" to perform the function "`UpdateBlockParms, MinX:MinY:MaxX:MaxY`"

20 Game Input Controls

The engine currently supports **Keyboard** and **PS3 Controller** when connected to the device, more support will be added for future controllers. Some detection code and user configuration options are required in a future update.

The following keys are used to control your Hero.

Keyboard Control	Action
Q, E, T, U, O, [, Cursor Left	Left
W, R, Y, I, P,], Cursor Right	Right
Z, X, C, V, B, N, M, Cursor Up	Jump
A, S, D, F, G	Pause the Game
H, J, K, L	Toggle In-Game Music ON/OFF
ENTER	Start or Go To Main Screen
F1 (On Piano Screen)	Put game engine in demo mode to show how each level can be completed without losing a life.
F1 (In Game)	If you have two or more lives, when activated (Cheat Mode not required) will lose a life, and show you how to complete the level to advance to the next, useful if you're stuck.
Numeric Keys	Cheat Mode Actions when Activated
Esc	Quit the game and go back to the Splash Screen

21 Cheat Codes

What retro style platform game wouldn't be complete without a cheat code?

Cheat mode can be activated by typing the following whilst playing any level.

21.1 Cheat Strings

- `6031769`
- `TYPEWRITER`
- `716486`
- `PPPENGUIN`

All codes activate the same cheat mode, which will be denoted by the BOOT on the screen.

Once Activated you can add extra lives or change the level.

21.2 Cheat – Extra Lives

Press the numeric keys `8, 9, 0` to add lives up to a maximum of 8.

21.3 Cheat – Change Level

The engine provides 40 familiar levels to play with. To change the Level, hold down the number `7` and then press and hold the following key combination and then let go of the `7` key. **(Maximum of 64 Levels)**

Key Combination	Level	Key Combination	Level	Key Combination	Level
7	1	75	17	76	33
71	2	715	18	716	34
72	3	725	19	726	35
712	4	7125	20	7126	36
73	5	735	21	736	37
713	6	7135	22	7136	38
723	7	7235	23	7236	39
7123	8	71235	24	71236	40
74	9	745	25		
714	10	7145	26		
724	11	7245	27		
7124	12	71245	28		
734	13	7345	29		
7134	14	71345	30		
7234	15	72345	31		
71234	16	712345	32		

22 Example GlobalConfig.cfg file

```
;
;                                                                   _                 _
;__      ____      ____      ____      _____  _ __ | | ___   _ _ __ (_)_  __  ___ ___  _ __ ___
;\ \ /\ / /\ \ /\ / /\ \ /\ / /\ \ /\ / / _ \| '_ \| |/ / | | | '_ \| \ \/ / / __/ _ \| '_ ` _ \
; \ V  V /  \ V  V /  \ V  V /  \ V  V / (_) | | | |   <| |_| | |_) | |>  < | (_| (_) | | | | | |
;  \_/\_/    \_/\_/    \_/\_(_)  \_/\_/ \___/|_| |_|_|\_\\__, | .__/|_/_/\_(_)___\___/|_| |_| |_|
;                                                        |___/|_|
;
;        www.wonkypix.com   and    manicmineragk.wordpress.com
;

;   Global Configuration File for the Wonkypix Manic Miner Engine
;   (C) Wonkypix.com
;
[NumberOfLives]
3


;
;   Set the Maximum Number of Levels available.
;       Remember this value must be correct!
;
[NumberOfLevels]
40

;
;   Set the name of the program loading
;
[ProgramName]
ManicMiner


;
;   Set Piano Screen Music
;
[PianoMusic]
Danube.ogg

;
;   Set Default Game Music
;
[DefaultLevelMusic]
InTheHallOfTheMountainKing.ogg

;
;   Set Demo Level Music
;
[DefaultDemoLevelMusic]
MountainKingOriginal.ogg


;
;   Define the Key Collection Sound
;
[CollectibleSound]
pick.wav


;
;   Define the Switch Sound
;
[DefaultSwitchSoundOn]
pick.wav

[DefaultSwitchSoundOff]
pick.wav

;
;   Define the Final Key Collection Sound
;
[FinalCollectibleSound]
pick.wav

;
;   Define the Default Level High Score Music
;
[DefaultHighScoreMusic]
EineKleineNachtMusik.ogg
```

```
;
;   Define the Default Extra Life Target Score
;       Set to 0 if no extra lives will be awarded
;
[TargetExtraLifeScore]
10000

;
;   Define the Default Additional Score in addition to [TargetExtraLifeScore]
;       for subsequent extra lives (0 if no additional lives earned)
;
[TargetExtraNextLifeScore]
10000
;
;   Define the Default Extra Life Target Score
;
[ExtraLifeSound]
ExtraLife.wav



;
; Default Item Points
;
[DefaultCollectablePoints]
100

;
; Default Miner Willy Fall Height (Blocks)
;
[DefaultHeroDeathHeight]
5


;
;   Define the Default End of Cavern 20 Music
;
[DefaultFinalCavernMusic]
CavernComplete.ogg


;
;       Default User Custom Sprite Sheet
;       Remember the .txt and .png must have the same name
;
[DefaultCustomSpriteSheet]
CustomSpriteSheet1.png

;
;       Want to add additional splash screens?
;
[SplashScreens]
CustomSpriteSheet1.png
Instructions.png
AdditionalCredits.png

;
; Allow Play Back of Levels (Yes or No)
;
[DisablePlaybackControl]
No
```

23 Example Level Configuration File

```
;
;                                                                    _                _
;__      ____      ____      ____      _____  _ __ | | ___   _ _ __ (_)_  __  ___ ___  _ __ ___
;\ \ /\ / /\ \ /\ / /\ \ /\ / /\ \ /\ / / _ \| '_ \| |/ / | | | '_ \| \ \/ / / __/ _ \| '_ ` _ \
; \ V  V /  \ V  V /  \ V  V /  \ V  V / (_) | | | |   <| |_| | |_) | |>  < | (_| (_) | | | | | |
;  \_/\_/    \_/\_/    \_/\_(_)  \_/\_/ \___/|_| |_|_|\_\\__, | .__/|_/_/\_(_)___\___/|_| |_| |_|
;                                                        |___/|_|
;
;        www.wonkypix.com   and    manicmineragk.wordpress.com
;

;   Level Configuration File
;   (C) Wonkypix.com
;
;
;Define The Level Name
;
[LevelName]
The Final Barrier
;
;  Set the Background Colour
;
[BackgroundColour]
BLACKCOLOUR

;
;  Set the Border Colour
;
[BorderColour]
DARKREDCOLOUR


;
;  Which of the 20 Sprite Sets do you want to use?
;  Starts (0-19)
;
[SpriteSet]
19

;
;  Define the Door Sprite To Use
;  (0-19)
;
[ExitBlock]
19

;
;  Now define the Level Data
;       0  = Empty Space
;       1  = Solid Platform
;       2  = Solid Platform
;       3  = Impassable Wall
;       4  = Collapsible Platform
;       5  = Hazard 1
;       6  = Hazard 2
;       7  = Conveyor Platform Moving Right To Left
;       C = Conveyor Platform Moving Left To Right
;       9  = Collectable Object
;       A = Willy's Exit/Door Position (Top Left Corner)
;       B = Special Control to detect Willy in the door
;       F = Special Block (Willy's Home Graphic)
;
[LevelData]
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFABF090000000F
FFFFFFFFFFFFFFFFFFFBBF000000009F
FFFFFFFFFFFFFFFFFFF00F000000000F
30000000000000000000000000000113
30000000000000000000000000000003
3CCCCCCCCCCCCCCCCCCCCCC000400003
30000000059050900509050000001003
31100000000000000000000000000003
30000110000000000000000000000003
30000000000000000000000000000003
31111111111111111111111111111113

;
; Sticky Conveyor
```

```
; If you want Willy to Stick to the Platform when landing
; and facing in the opposite direction and the player
; is trying to move in the opposite direction
; Set Value to 1 if you want Willy not to Move
; Set Value to 0 if you allow Willy to move against the flow
[StickyConveyor]
1

;
;   Define the Actors on the Caverns
;
;   ActorName, Start Block X, Start Block Y,
;                           Initial Direction,
;                           Min Block X, Min Block Y,
;                           Max Block X, Max Block Y,
;                           dx (pixels), dy (pixels),
;                           tick count
;                           leftFrame, rightFrame, UpFrame, DownFrame, maxFrames, Frames Per Animation Seg
;
[Actors]
YellowCamera.png,   7, 13, 1,  7, 13,  22, 13, 16, 0,  4, 4, 1, 8, 8, 4, 4
Eye.png,           24,  6, 2, 24,  5,  24, 13,  0, 2,  4, 4, 1, 8, 8, 4, 4

;
;   Player Start Block Position
;
;
;   Actor Filename, Start Block X, Start Block Y, Initial Direction, DX (Pixels), DY (Pixels), Tick Count, Segments

[HeroStart]
MinerWilly.png,28,13,16,16,2,4,16

;
; Trigger Caverns Complete Sequence (Graphic Must be in Main Sprite Sheet)
;
;       Graphic Name, Block X, Block Y, Manic X, Manic Y, Frame
[SwordFish]
SwordFish.png   , 19, 5, 19, 2, 4

;
;   End of Cavern 20 Music (Needs Swordfish)
;
[FinalCavernMusic]
CavernComplete.ogg


;
; Replay Data to demonstrate how the level can be completed
; Must be the last Tag
;
[ReplayData]
A20h20a800=0<=0H=0A>00B0<B0PB0IE08G0DG0XG01I0MJ0aJ0mK0IL00M0jM0NN0bN08Q0RW0^X0:Y0hY0:Z0NZ0R[0`[0Q\0P
]0I^0X^0B_00`09`0Hb01j0P01<21X21Q41E71Q71e81I91P91R:1^;12<18=14D1XD1
```

Change Control Record

Date	Author	Version	Change Reference
5th May 2020	Jason Brooks	V0.1 Draft	Initial Version for Review.
6th July 2020	Jason Brooks	V0.2 Draft	Updated based on feedback from Bob

What's New?

This is the initial release manual for the Config Driven 2D Gaming Engine aka Manic Miner Engine, and being made available from Engine Version 1.6a.

Copyright

DOCUMENT CONTROL STATEMENT

Printed or Electronic of this document are uncontrolled.