

# McPixel CREATOR'S GUIDE

This guide will teach you how to use McPixel level creator, how to create and submit levels and reveal few secrets behind the way the levels are beign created, stored, and handled.

## Contents

- **Chapter 1 - Quick Start**
- **Chapter 2 - The Creator**
- **Chapter 3 - The Image**
- **Chapter 4 - The Script**
- **Chapter 5 - Graphics Gale**
- **Chapter 6 - Submitting**
- **Appendix A - Console errors**
- **Appendix B - Legal shit**
- **Appendix C - Graphics Gale**

# 1. Quick start

Ok, for a kickstart I'm going to show you how to get a level running in the creator and how does the thing work. First of all you will need the **Creator's starter package**



**Download McPixel creator's starter package**

[http://mcpixel.net/mcpixel\\_creator\\_pack.zip](http://mcpixel.net/mcpixel_creator_pack.zip)

Let me quickly enumerate what's **in this package**...

- ▀ Source files of all original 6 levels
- ▀ Scripts for these
- ▀ McPixel spritesheet
- ▀ Several backgrounds
- ▀ Some random graphics (bombs, items, explosions)
- ▀ Pair of brand new shoes
- ▀ This guide

This is going to give you a kickstart for creating the levels. It might sound like a lot of work overall but it's actually not that much. Let's move on! Now **launch McPixel** and get into the **create mode**.



This is how it looks like. I will elaborate on all the buttons features later on, for now, punch the **LOAD IMAGE** button. It will open a file browser dialogue. Select one of the levels from the creator's pack, for this instance, let's say **lvl\_tram.png**

The level starts immediately and you will see the level intro screen!



You can walk around and interact with stuff, but for now it will not trigger any actions. To do that we need the script file. Punch the **LOAD SCRIPT** button and choose the level's corresponding script file, for this instance, **lvl\_tram.txt**.

Now you can actually trigger actions by interacting with stuff! If you are amazed by this magic tricks of mine, please **take a look at the png and txt** files to see how little it takes to create a level!

If you want to delve into how all this works, you will need to wait! There are still buttons features left for me to describe out there!

## 2. The Creator

What I call The Creator is actually more of a debugger. You can't draw the actual level or edit scripts from inside this tool. This is because levels are actually handmade pixel art that you need to draw in your favourite graphics editor and scripts are just text files that you write in your favourite text editor!

The Creator has a number of buttons features that will help you create levels for McPixel as well as a level submission tool. So that if you are satisfied with the level, you can immediately submit it. After submission I'll check it out and if I get a full six-pack of these, I will publish it immediately!

Ok, let me write what kind of buttons features are included in the creator.

### LOAD IMAGE

Punching this button feature will open a file browser dialogue, where you can pick a level image. After it is picked, the level will start immediately. If a level is already loaded, it will restart the level after it loads. You can quickly reload stuff when you are making changes using this.

### LOAD SCRIPT

This does the same to the script. You have a file browser and you can load or reload your script at any time. Doing so restarts the level as well.

### DEBUG INFO

Displays useful information about items in the level. Every item is outlined and the ID and frame number of each one of them is displayed. It's useful if you have trouble with items or you want to see the IDs of the items to write the script!



# CONSOLE

Console will log any messages from the creator so if something is wrong, you should take a look at it. It will also log stuff if everything is fine so you should read it to rejoice about that as well! There is a detailed description of the console and error messages later.



# QUIT TO MENU

Guess what this does!

# SUBMIT LEVEL

When you are satisfied with your work, punch this button feature to submit the level! There is a form to fill before you can actually submit the level.



- **AUTHOR** - Type your name here so that you can be credited for your work!
- **EMAIL** - Type your e-mail address. I will not be sending any spam. It's just for contact if there is anything wrong with the level that I'd need to discuss before publishing it!
- **LEVEL TITLE** - Name your level!
- **NOTES** - If there is anything you'd like to share, express or vent, type it there!

## UNLOAD

You can unload the level to admire the NO LEVEL LOADED message some more! Actually this button feature is useful if you are loading a different level without the script. By default the old script will still be used, so you need to unload it first.

## RESET TIME

Sets the countdown timer to 60 seconds. If you are a loser that can't do stuff quickly enough, you can have more time to admire your level.

## RESTART

Restarts the level immediately.

## LOSE

Plays the losing animation immediately. Useful if you clicked the RESET TIME button.

## WIN

Enables you to win anything in life. Lotteries, cars, stuff. Also, this is not true. Actually it plays the winning animation. Useful when you don't have a script yet.

## HELP

Opens this help! Neat, huh?

# 3. The Image

Ok, all this I explained is quite useless without having basic knowledge about how the levels actually work. It's actually pretty simple, so let me delve into that!

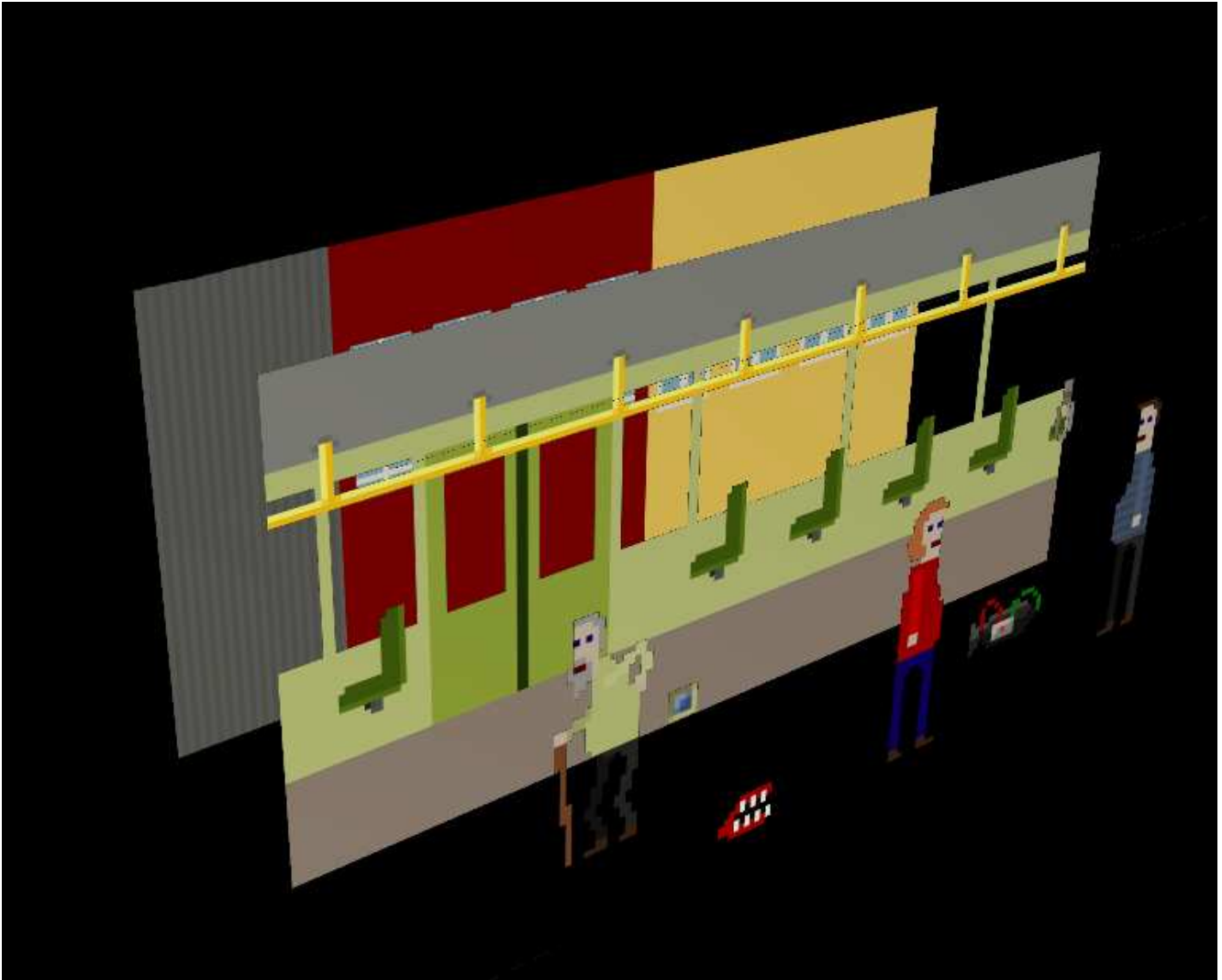
The actual level consists of two parts. First is the level image, and the second is the script. This part covers the former.



This is it. When you load such image into the creator you are ready to play it. Actually, the most important part is the first two rows. And these are enough to run the level. There are, however few rules that the image must meet.

- The level image works like an animation spritesheet
- Each frame must be exactly 160x80 pixels in size
- The width of the image must be at least 4 frames (640 pixels)
- There have to be at least 10 frames for the level to run
- The level needs to be a PNG file
- Transparency colour or alpha channel must be used to account for transparent areas.

The level is created out of this image by layering several frames on top on each other. During the play there are 3 layers.

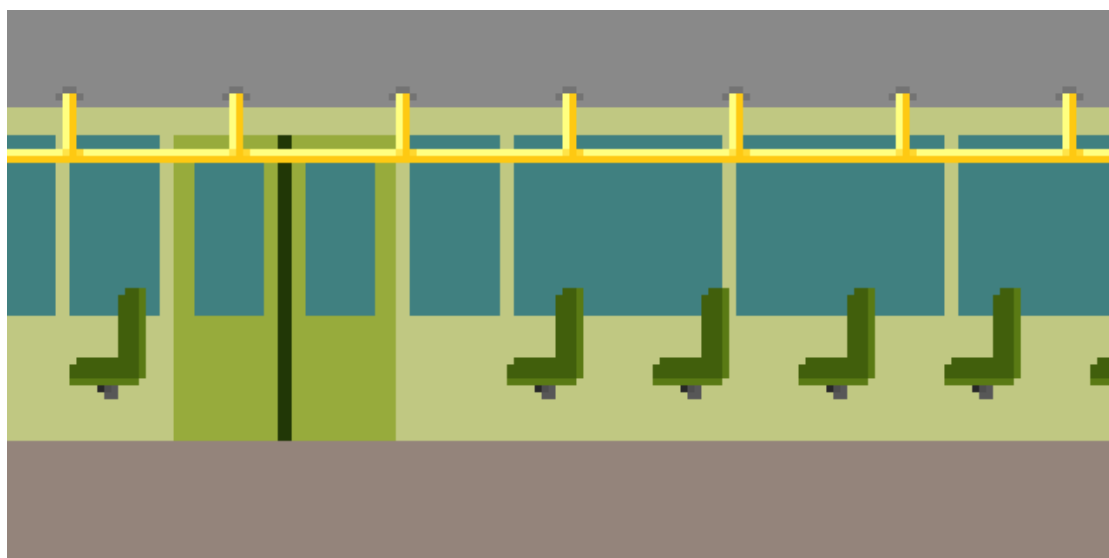


- **Scrolling background layer** - This is the bottommost layer. It is a background that continuously scrolls in one direction. You can set the scrolling direction and speed yourself.
- **Static foreground layer** - This is the actual scene around which McPixel walks.
- **Animated sprite layer** - This layer consists of items, people and of course, McPixel himself.

These layers are obviously defined by separate frames from the level image sheet. The order in which these appear is fixed. Let me elaborate on each of the frames and their meaning. Oh, and as the frame numbering system is zero-based (the first frame is frame 0) I am going to use that numeration here.



## FRAME 0 - The foreground



This is the foreground. It's completely static, meaning it does neither animate nor scroll. While drawing this you should keep in mind the area on which McPixel can walk.



The walk area is at 64px-80px, the bottommost 16 pixels. McPixel under no circumstances will go higher than that. However, when interacting with items, he might step out of this area sideways or below. You should keep in mind to prevent this. Moreover, you should keep a threshold. So that the actual ground is drawn from at least 62px. This way McPixel won't appear as plastered onto the wall when he is in the highest possible spot.

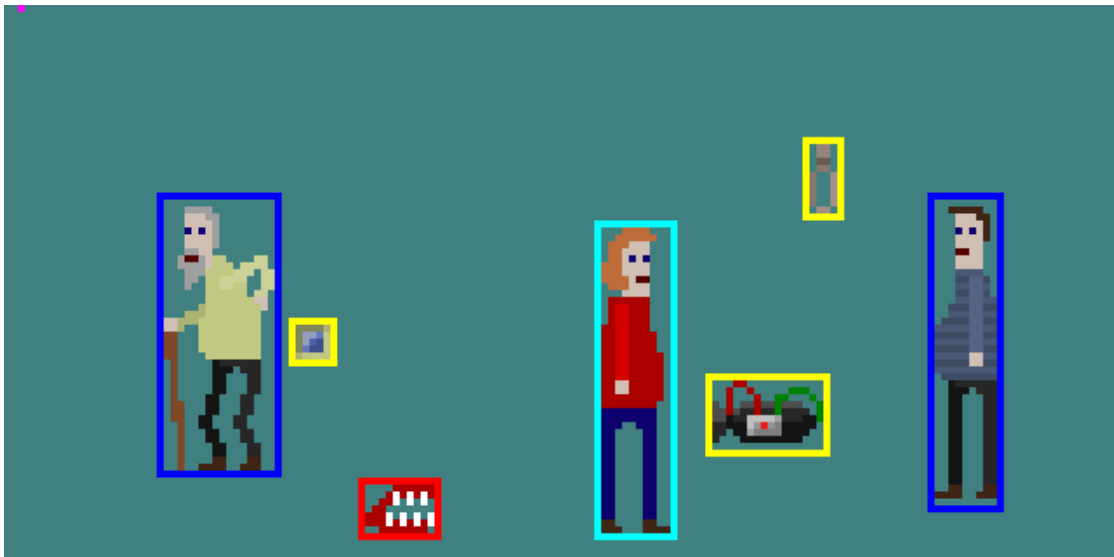
## FRAME 1 - The background



This is the scrolling background. You can control which way it will scroll. You should keep in mind that the background should tile (be repeatable) in the direction of scrolling. The slowest scroll setting is pretty fast to emphasise the fast-paced nature of the game. I will explain how to control scrolling in a moment. Also, you can disable scrolling at all.

Most of the times only parts of the background are visible, and you should keep that in mind. As you can see on the picture, some parts of it are invisible during play and can be disregarded.

## FRAMES 2-3 - The items



This is the most important part of the level image sheet. It defines the items that appear in the level, scrolling direction, speed and McPixel starting position. There are two frames for that purpose, because items can be animated. The position of items is determined via encapsulating them in boxes of a specific colour. Let me explain meaning of colours.

**Magenta #ff00ff rgb(255,0,255)** - One pixel in this colour indicates the direction and speed of scrolling. In this instance, the background (from frame 1) will constantly scroll left. If you want to scroll it right, place it near the right edge of the screen. If you want it to scroll up, place it on the left vertical edge near the top edge. if you want to scroll down, place it near the bottom edge. If you want it to look eye-bleedingly broken, place the dot near the center of the image.

**Yellow #ffff00 rgb(255,255,0)** - A box drawn in this colour around a graphic indicates an item. Everything inside excluding the frame will be converted to a separate sprite and used as an item. Yellow indicates a static item, one that you cannot grab and carry around. When interacting with this item McPixel will either grab it with hand or low-kick it depending on its position (explained later). There must be a 1 pixel gap between items, meaning that boxes cannot touch each other.

**Red #ff0000 rgb(255,0,0)** - Red indicates a pickable item, one you can pick and carry around. The actions defined in the script can, and should be different depending whether McPixel is holding an item or not. There should be only one pickable item in a level for simplicity. Having more than one is not supported from the script, so that even if there are two, both will launch one action.

**Blue #0000ff rgb(0,0,255)** - A man, or NPC if you want to put it that way. The only actual difference between static (yellow) items and NPCs is that default interaction with an NPC is kicking his balls. Feel free to abuse that and have McPixel kick balls of stuff other than people.

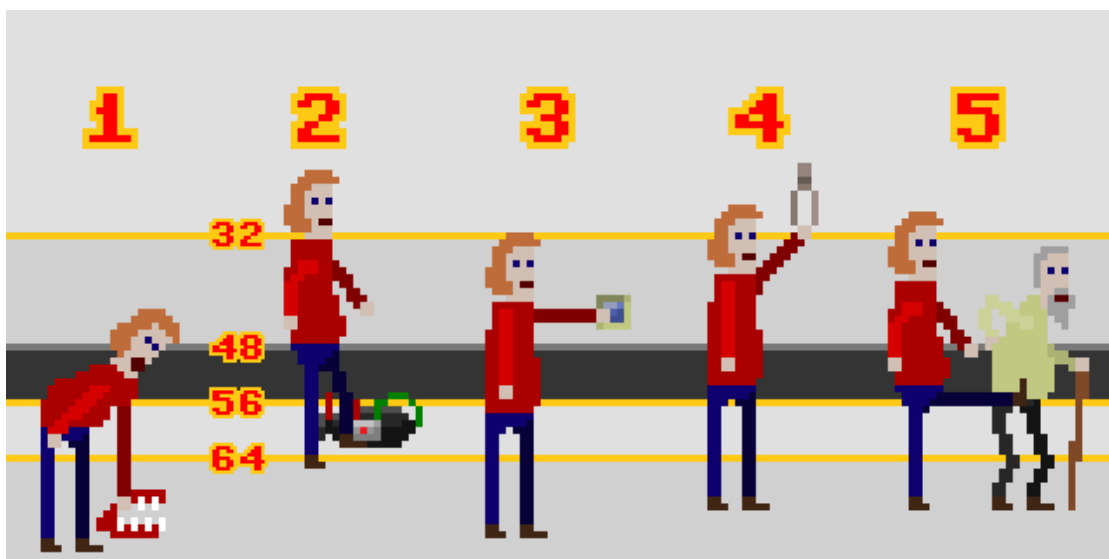
**Green #00ff00 rgb(0,255,0)** - Green works the same way as yellow. The difference is that it indicates a background item. By default McPixel can go behind and in front of stuff depending on his position. however, if you use this colour, McPixel will always appear in front of this item. To see how it can be useful, take a look at The Space level example from the package. There is a floor latch boxed with yellow, which means McPixel can go behind the latch. Change it to green to see how it affects it! This is useless for stuff that is on the wall, since it will always be behind McPixel anyways.

**Cyan #00ffff rgb(0,255,255)** - This indicates where McPixel is standing in the beginning. It is enough to have a cyan dot to indicate that, so feel free to do that if you lack free place on the level. However, I suppose it looks cleaner when you paste McPixel and box him with cyan.

These colours are exclusive for the purpose of describing level contents. So if you want to draw something yellow, use a nearby colour to do that (#fefe00 rgb(254,254,0)). This way it is almost unnoticable. Moreover, everyhting that is not framed, will be discarded, so that if you want to temporarily remove an item from the stage, it is enough to delete the frame surrounding it.

The default action McPixel takes when interaction with an item is determined by its position. The position of item is its center. If you are unsure about that position, enable the debug info and a crosshair will appear in the middle of each item. If you want McPixel to approach item from palce other than its center, you can easily fake achieve this by extending the item frame in a direction. Browse the tutorial levels to see numerous examples of doing that.

Now, let me enumerate the interaction zones, and how McPixel does something different to an item depending on its position.



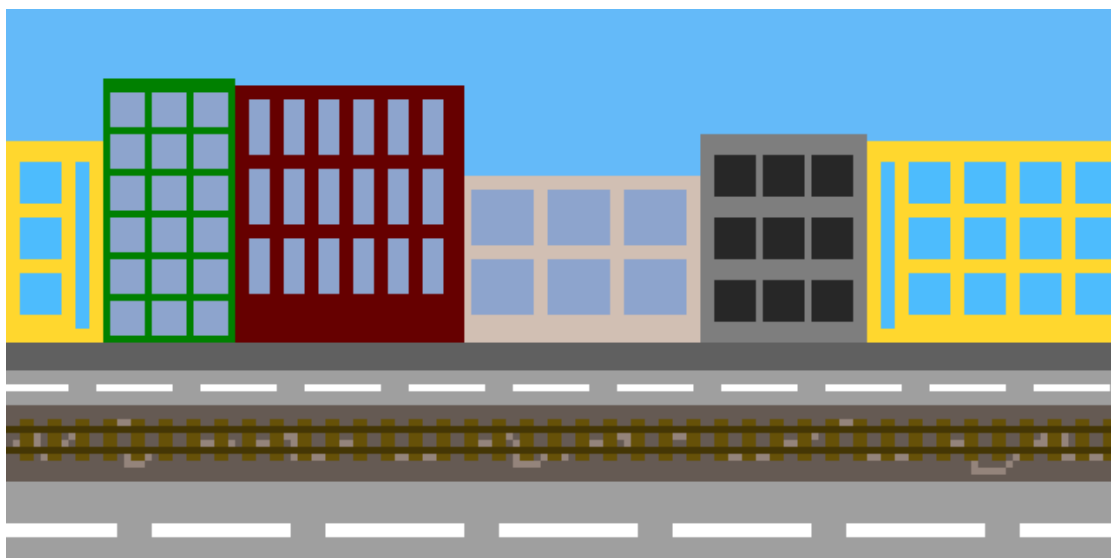
- 1. **Bend zone (64px-80px)** - In this zone, McPixel will bend over to reach for the item. This is most preferable for pickable items.
- 2. **Lowkick zone (56px-64px)** - McPixel will lowkick everything in this zone. It's a bad idea to put pickable items here.
- 3. **Punch zone (32px-56px)** - McPixel will reach in front of him for the item.
  - !!! **DEADZONE !!! (48px-56px)** - This is a part of Punch zone, where McPixel will slightly exit the screen to reach the item (his feet will be below the screen). You should not put anything there.
- 4. **Hail zone (0px-32px)** - McPixel will reach up for the item placed here.
- 5. **Ballkick zone (64px-80px)** - McPixel will align his feet position with that of the NPC and ballkick him. Also, the crosshair in the creator will point to NPCs balls, instead of the sprite center, so that you can aim your hits well. Moreover, considering how McPixel has to align his feet with NPCs feet, the NPC feet have to stay inside the walk zone.

When drawing the level, you should first create the 10 base frames, and before drawing the cutscenes, you have to load it, and see where exactly McPixel is standing when interacting with the item to draw the first frame. If you mismatch his position by as little as one pixel, there will be a visible stutter, and you will have to redraw your animation!

To conclude elaboration on this part, let me enumerate several guidelines you have to keep in mind when drawing this part.

- The magenta scroll indicator is compulsory and you only can have one such pixel.
- You need to have neither less nor more than one McPixel. Actually, you can have more, but only one can be the player character, framed with cyan.
- You can have no items at all in your level and it will work just fine!
- I totally have no idea what will happen if you screw something up here, so please don't do it.

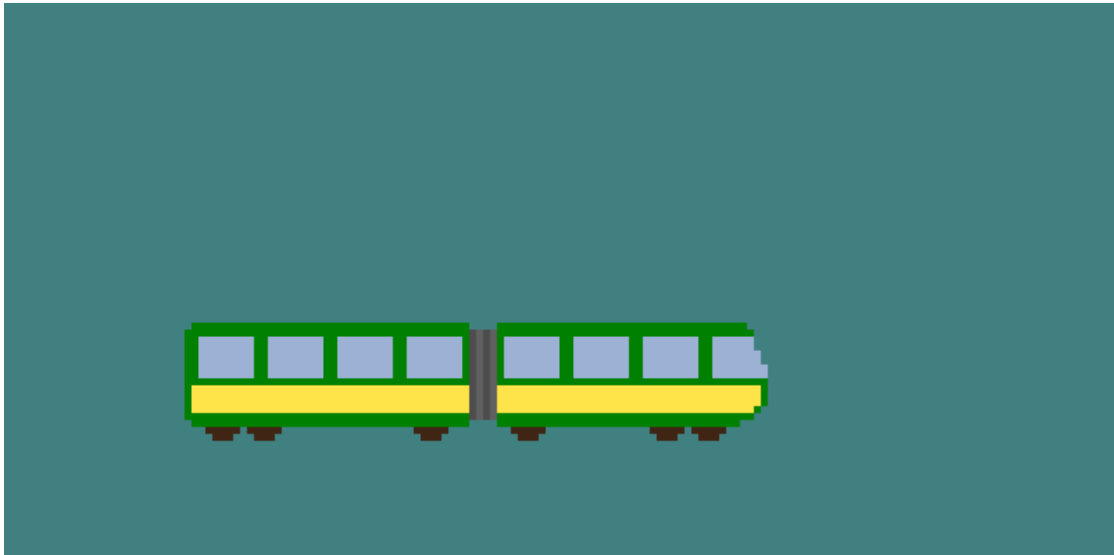
## FRAME 4 - Cutscene background



This is the same as Frame 1, but applies to the cutscene (the one that displays McPixel logo). It's a scrollable background and the scrolling speed and direction is the same as when playing the level and defined in the item frame.

This and all (of the 10) subsequent frames are used for the cutscene. They should provide a view from outside / away of the actual level scene.

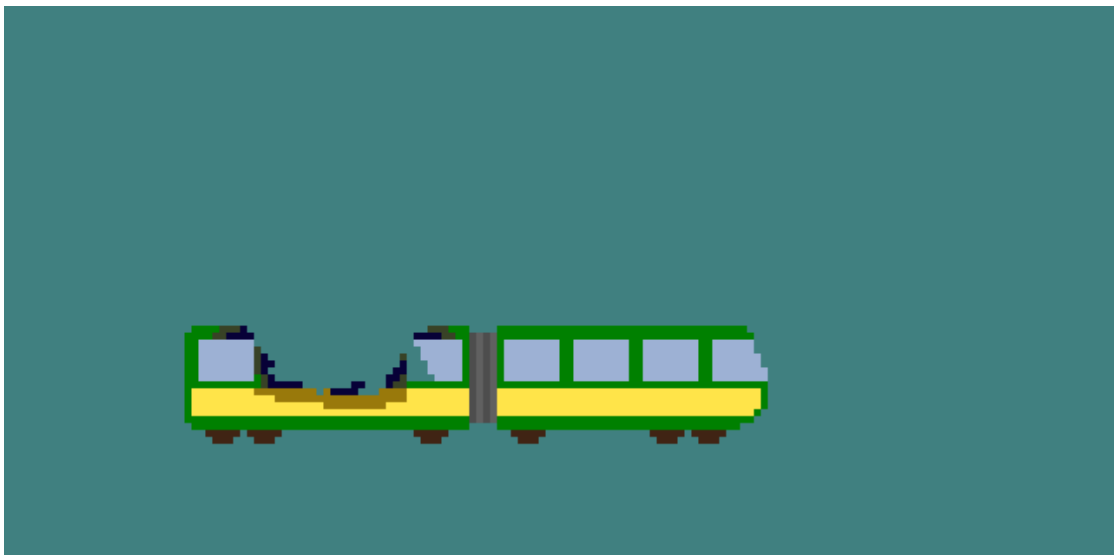
## FRAMES 5-6 - 'Good' cutscene



It's 'good' in quotes, since it's used all over, not only for the good ending. Basically it's a static foreground of the cutscene in two frames. As you can see above, the frames are not toggling, but more of stuttering between each other. Places it is used in:

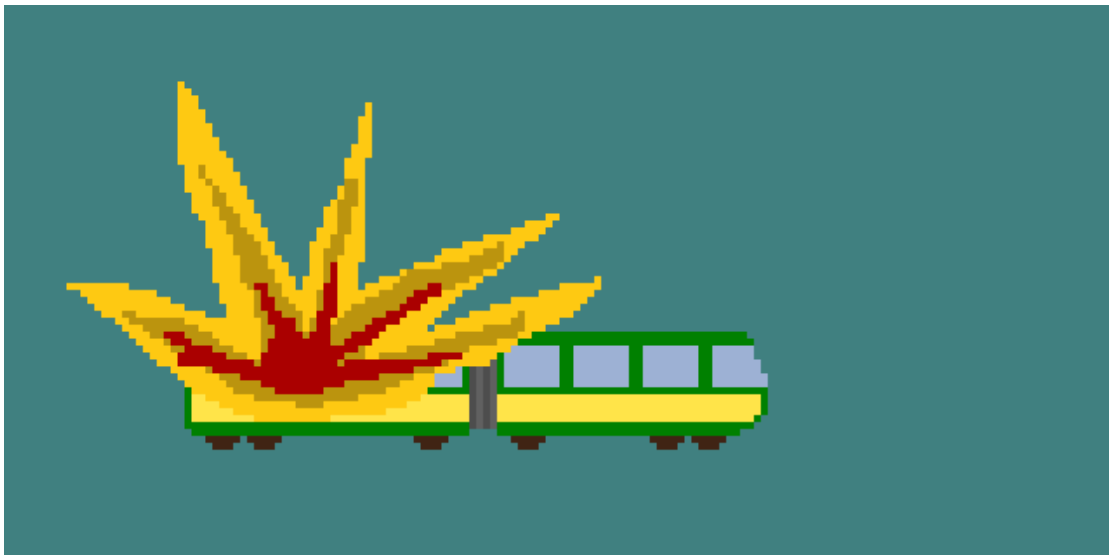
- **The intro** - This animation plays before you get to play the level. It plays similarly to the one above
- **Win cutscene** - The winning cutscene is basically the same as the intro.
- **Lose cutscene** - This also plays in the losing cutscene, before it blows up

## FRAMES 7-8 - 'Bad' cutscene



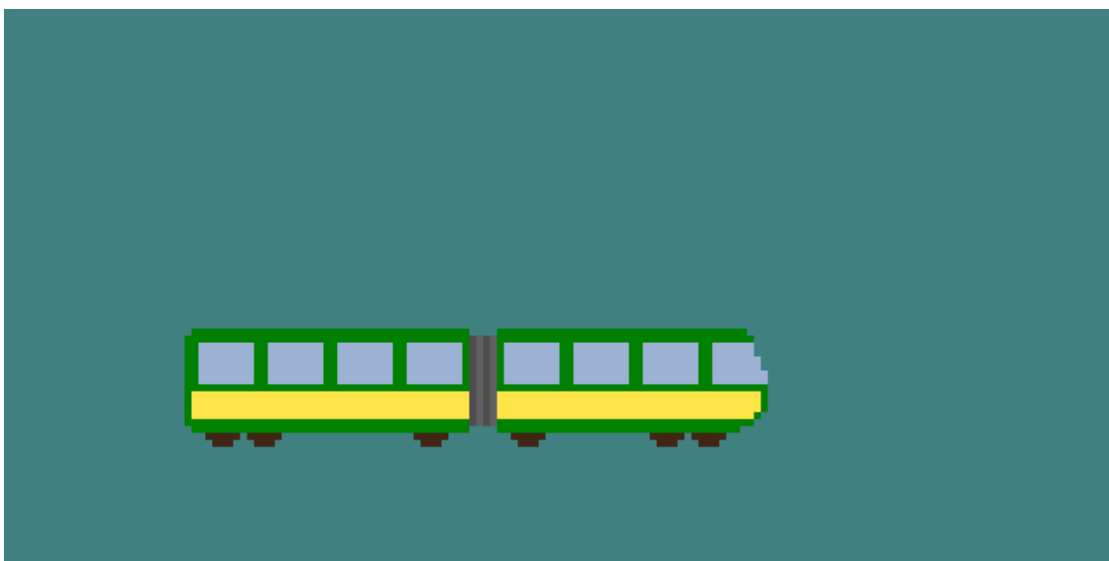
In a nutshell, these are the same that frames 6-7, but after the explosion. It animates the same way that the above one. Frame 7 should correspond to frame 5 and frame 8 to frame 6.

## FRAME 9 - Boom!



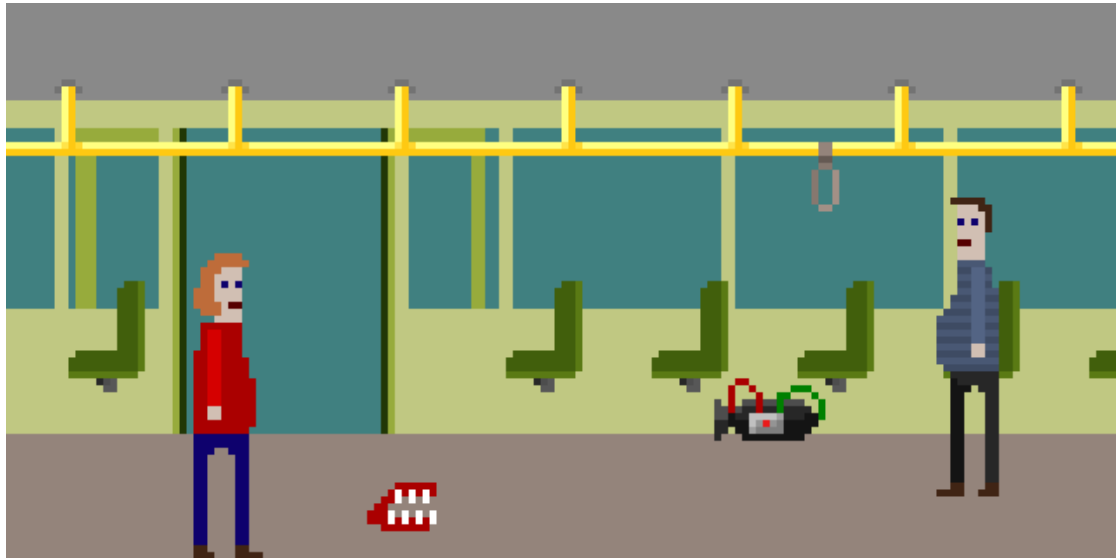
This is the most important part of everyone's lives - everything explodes. This frame should correspond to frame 6, not frame 5, since it's being played after frame 5 and before frame 7.

## FRAMES 5-6-7-8-9 combined



This is the 'bad ending' when you lose and stuff blows up. It uses all of the cutscene frames.

## FRAMES 10+ - Endings



Arrangement of any of the following frames is up to you. These are used for the animations of endings and their playing order is determined via the script. These can be sparse, meaning that you can have unused frames in there. Also, you can have a custom scrolling background for your ending, which you can place anywhere and access via the script.

Oh man... so much about the script, I better move onto it now!

## 4. The script

```
level 1
item 0 | hold | nowin | .1 | 1 | 22, 23, 22, 23, 22, 23, 22
item 0 | nohold | nowin | .1 | 1 | 11, 10, 11, 10, 11, 10, 11, 10
item 2 | hold | win | .1 | 1 | 24, 24, 25, 25, 24, 24, 25, 25,
..... 26, 26, 27, 27, 26, 26, 27, 27,
..... 28, 28, 29, 29, 28, 28, 29, 29,
..... 30, 30, 31, 31, 30, 30, 31, 31,
..... 32, 33, 34, 35, 36, 37, 38, 39,
..... 39, 40, 39, 40
item 3 | nohold | nowin | .3 | 1 | 12, 13, 14, 15, 16, 17, 18, 19, 20,
..... 21, 21, 21, 21, 21, 21, 21, 21, 21,
..... 21, 21, 21, 21, 21, 21, 21, 21, 21,
..... 21, 21, 21, 21, 21, 21, 21
item 4 | nohold | nowin | .1 | 1 |
item 4 | hold | nowin | .1 | 1 |
thanks
```

This is how the script looks like. It defines animation sequences that are triggered by interacting with each item. the script has 3 parts, let me describe them.

- **The beginning** - The first line. It has to say 'level' and then there's a number. I forgot what the number is for so just pick a random number to place there yourself. Oh, and the number is compulsory
- **The middle** - There are actual action definitions, Each starts with new line but can take more than one line. Each need to start with 'item' keyword.
- **The end** - In the end you say 'thanks'. It doesn't work if you don't do that.

The most important part is the middle where you actually define the animations. The syntax of each definition is as follows.

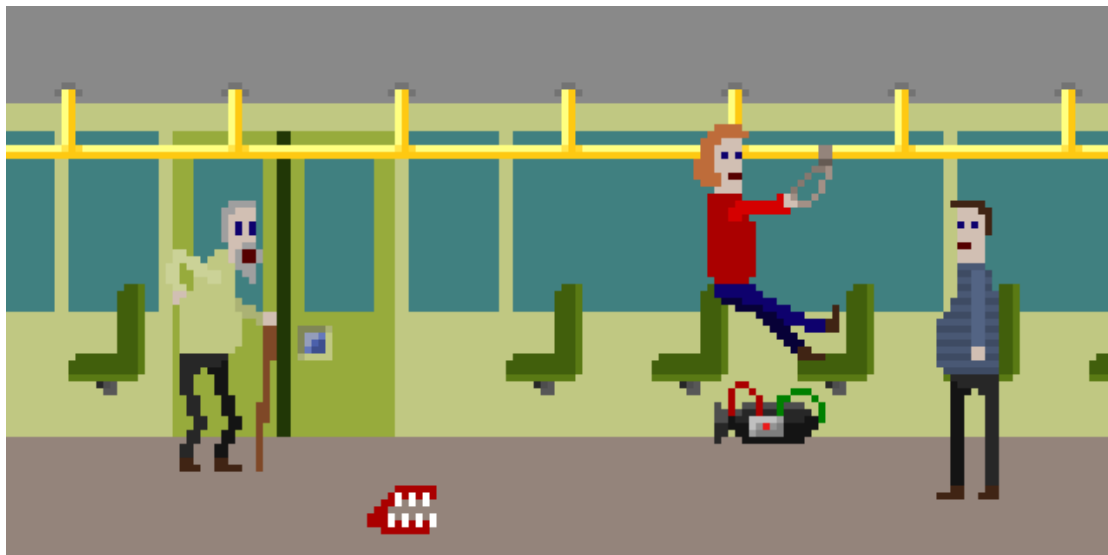
```
item [id] | [hold/nohold] | [win/nowin] | [frame_rate] | [bg_frame] | [frames]
```

- **item [id]** - Declares which item is this about.
- **[hold/nohold]** - Whether McPixel should be holding a pickable item to trigger this.
- **[win/nowin]** - Is this the winning solution?
  - **[frame\_rate]** - Frame rate in weird format. 0.1 is my default and a good place to start. 0.2 is two times faster and good for smooth animations such as falling. If you want variable frame rate, Use 0.2 and duplicate frames that should be slower.
- **[bg\_frame]** - Scrolling background frame. If you don't have one, type 1 to use the level's default one (at frame 1)
- **[frames]** - List of comma-delimited frame numbers for this animation. Frame numbers are 0-based, so that the first frame is frame 0, and if there are 20 frames, the last frame is frame 19.

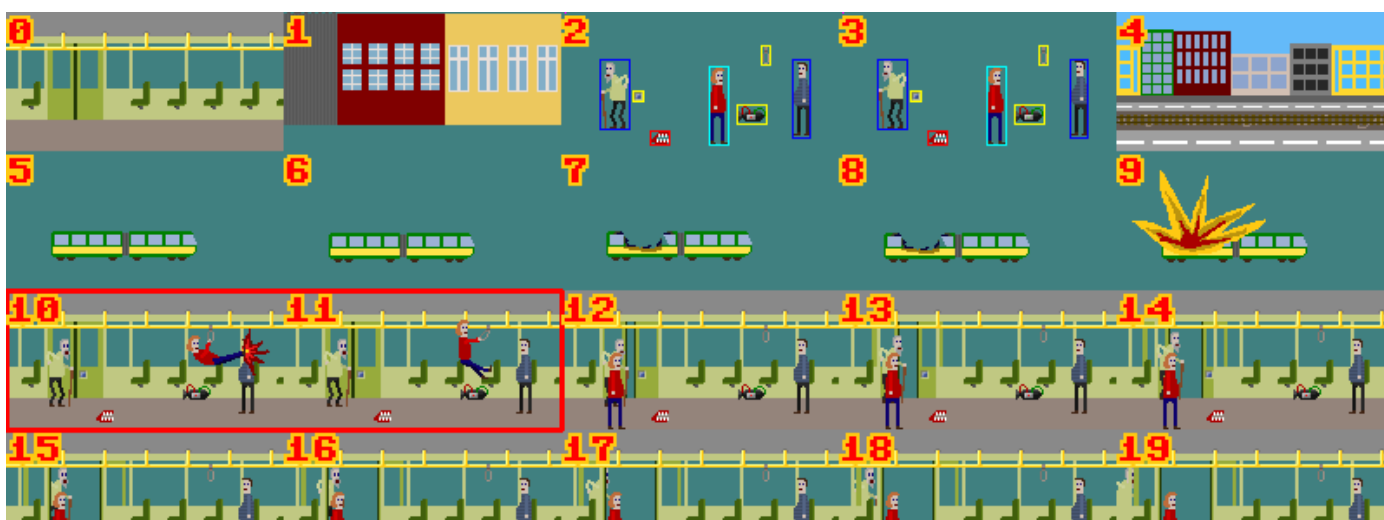
If there are any errors with your script file, you will be notified via the console. Check the console after loading the script to make sure that everything went fine!



Let me show you an ending animation, the script line that triggers this animation and position of this animation in the level image sheet.



```
item 0 | nohold | nowin | .1 | . 1 | 11, 10, 11, 10, 11, 10, 11, 10
```



This is basically how it works. I guess writing the scripts is not that hard! But since there are some guidelines to follow while writing them, let me enumerate these.

- You can have two outcomes triggered by one item. One while holding another pickable item and one without it.
- The same animation cannot run for holding and not holding an item, since the pickable item location and therefore animation would be different for these cases.
- There can be no animation frames at all, this way the ending cutscene will play immediately
- There's no way to control more than one pickable item from within the script
- You can have more than one winning solution, but that doesn't mean you should actually do that.
- Items are numbered linefirst, meaning that item with ID 0 is the topmost and leftmost one. Use the DEBUG INFO to see the IDs if you have any concerns.
- Pickable items do not trigger animations themselves. They can only be used on other items.

This is it about the scripts. Let's move on...

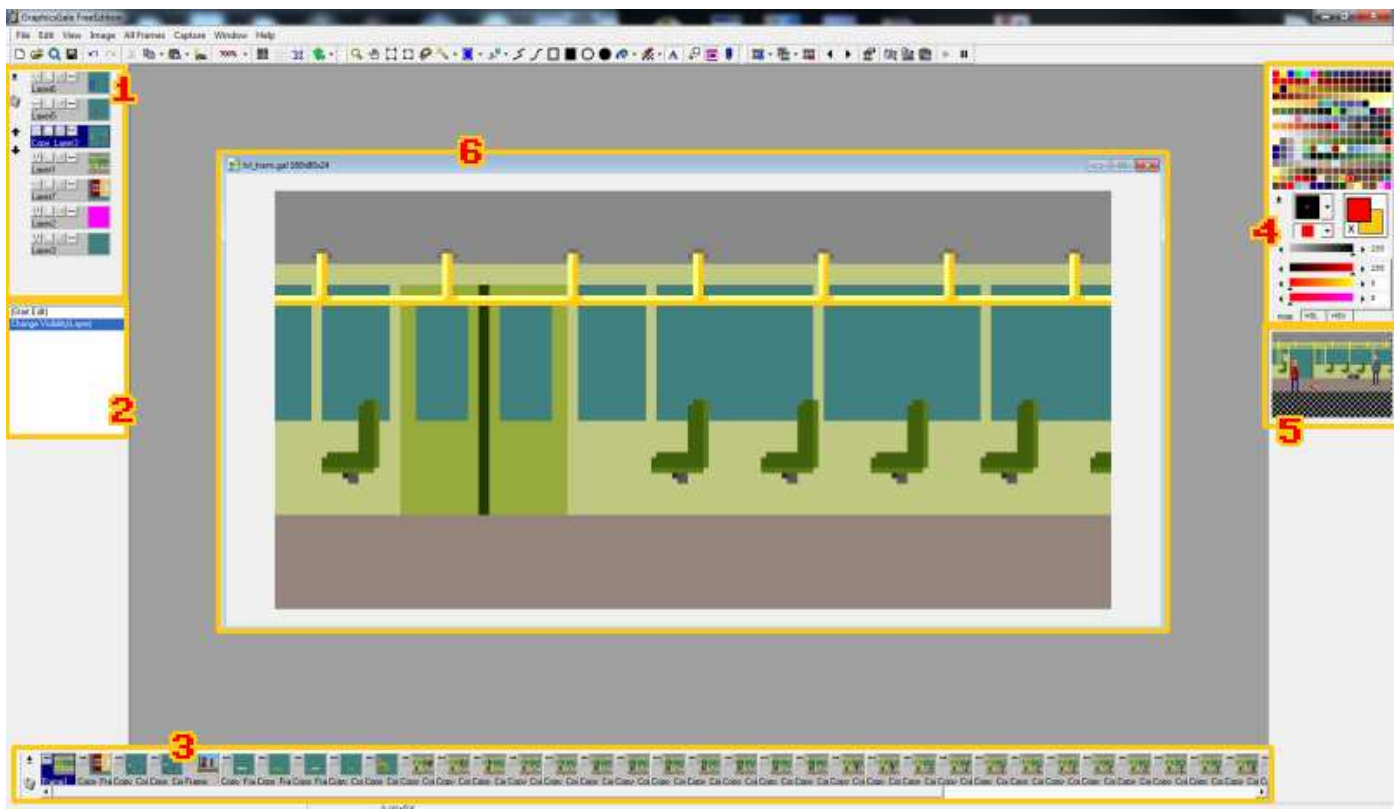
# 5. Graphics Gale

Graphics Gale is by far my favourite drawing tool. I use it in conjunction with Irfan View and XnView. I use the former to achieve transparency in PNGs quickly and the latter to add effects to images. Let me point you to download spots for these.

- **Graphics Gale** - <http://www.humanbalance.net/gale/us/> - Make sure to get the 2.0 beta version!
- **Irfan View** - <http://www.irfanview.com/> - You probably have this already!
- **XnView** - <http://www.xnview.com/> - It's lightweight and hosts Photoshop filters!

Of course you can have your own favourite graphics app, like Photoshop, Deluxe Paint or MS Paint even. It's not needed to be using Graphics Gale, but since the source files for the levels are in .gal format, you should get it just to examine them.

And since this is not a pixel art or a Graphics Gale tutorial, I'm not going to delve into details of those much. But still I will provide few hints and an overview of how the application works, since it's crucial to understanding this chapter. If you are running Mac OS or Linux, you can easily launch Graphics Gale using, correspondingly, Wine or WineBottler. Let me tell you about the basic features of Graphics Gale. Launch the application, and load lvl\_tram.gal file.



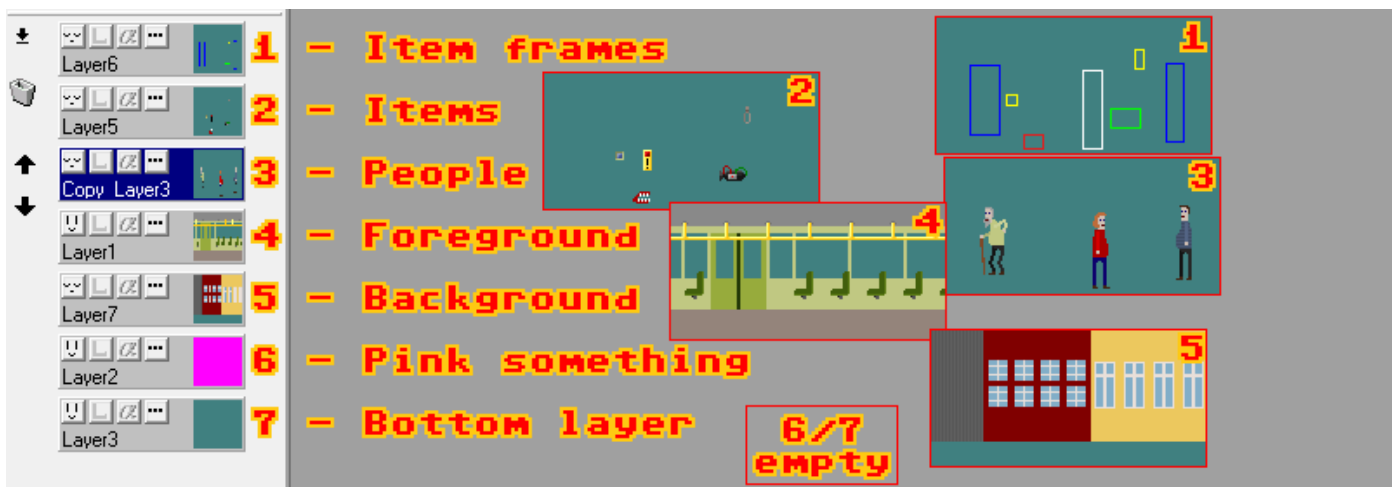
This is NOT what you're going to see immediately, you need to do two things first.

- Use the magnifying glass tool to zoom into the level image as much as you can!
- Then, click the ↓ (eject-shaped) button near the palette on the right, and load palette.pal file from the creator's package.

If your layout looks totally different from what you can see here, it means you got the wrong version (the 2.0 beta is the one you should get). But you can easily drag stuff around and rearrange it to look like this. Now let me tell you about the basic features of this app.

- 1. **Layers** - The image consists of several layers. You can enable and disable them. Each layer can have a separate transparent colour or an alpha (transcparency) channel.
- 2. **Actions** - List of stuff you did. If you want to undo/redo, you can take a look at these and see what you did wrong.
- 3. **Frames** - There's quite a number of these in the level source files. Each frame can have a separate set of layers.
- 4. **Palette** - Colours for you to pick. The palette file you loaded from the package is my working palette, and it constantly changes. You should feel free to be changing your palette too.
- 5. **Preview** - Image preview. You can control it's zoom by right clicking and choose whether it's animated using play/pause buttons on the toolbar.
- 6. **Image** - This is the image you are editing. You can have multiple of these open simultaneously.

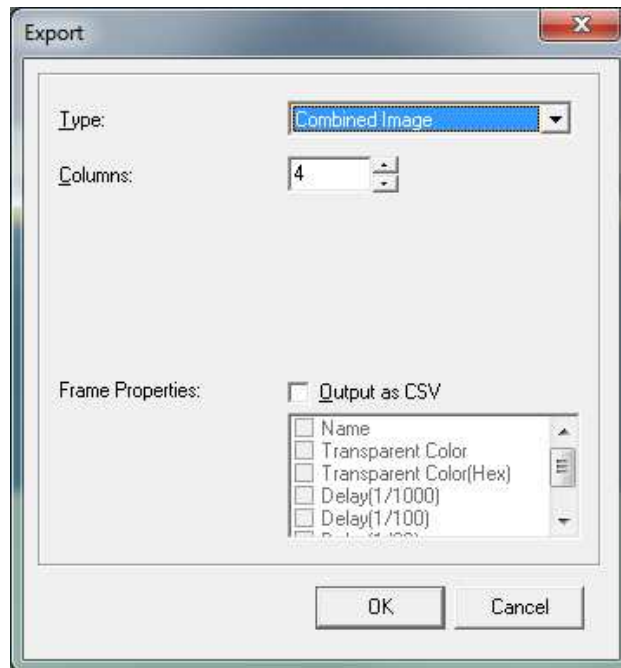
This is the basic layout for the application. If you want to delve into it some more, I will describe some features and techniques to help you get used to this in an Appendix. For now, I would like to focus on how to keep your levels arranged in layers and frames.



This is the layer layout for frame 0. The subsequent frames are just a copy of this frame with different layers enabled. I would like you to look at the bottommost layer. As you already noticed I am not using magenta (magic pink) for transparency. One of the reasons is that it hurts your eyes to look at it for long time. The other is that Graphics Gale would substitute the default transparent colour with white. The bottommost layer is holding the transparent colour, and has no transparency enabled. This way it's easy to pick the transparent colour (right-click is color picker) from the image.

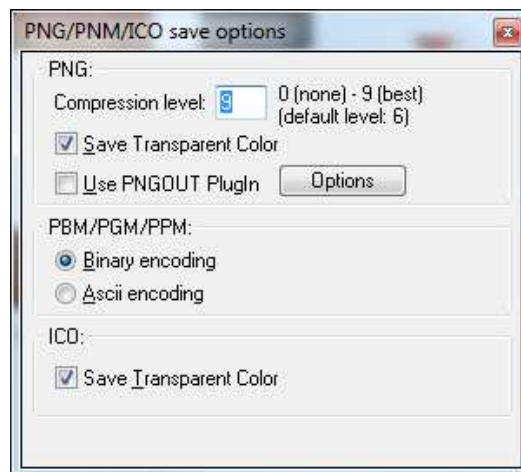
Having you layers arranged this way is also helpful for animating the ending. Take a look at any animation frame to see that the layers are basically in the same arrangement that frame 0. The same goes for the cutscene frames (frames 6 through 9) where all of the ending and intro animation is on one frame, divided by layers and onle some are enabled each frame. Take your time to browse the source files and see how they are arranged.

You might want to make changes to the level, but keep in mind, that if you change anything in the base level, the animations will become obsolete and you will have to alter all of the animation frames to match the changes. So always remember to create the base level first, and only after you are sure that everything is fine, proceed with creation of the animations. So, to make the level into a PNG image sheet that you can actually load into the editor, you need the Export tool from the File menu.



Always keep the Columns value at 4 or more! Remember that the level needs at least that to even load into the creator! For now, just make or not make some changes and export as a BMP file.

BMP format doesn't support transparency, so now you have to load the file into Irfan View, and save it back as PNG. While doing that, enable the Options dialogue and check Save Transparent Color checkbox.



You will be asked to pick the transparent colour from the image with your mouse. Just click a place with the colour that should be transparency, and you now have a PNG file that you can load into the creator.

## 6. Submitting your level

When you are done and satisfied with your work, it's time to submit your level. To do this simply punch SUBMIT LEVEL button in the creator. If your level is OK, you will see the submission form immediately, otherwise, you may encounter an error message.

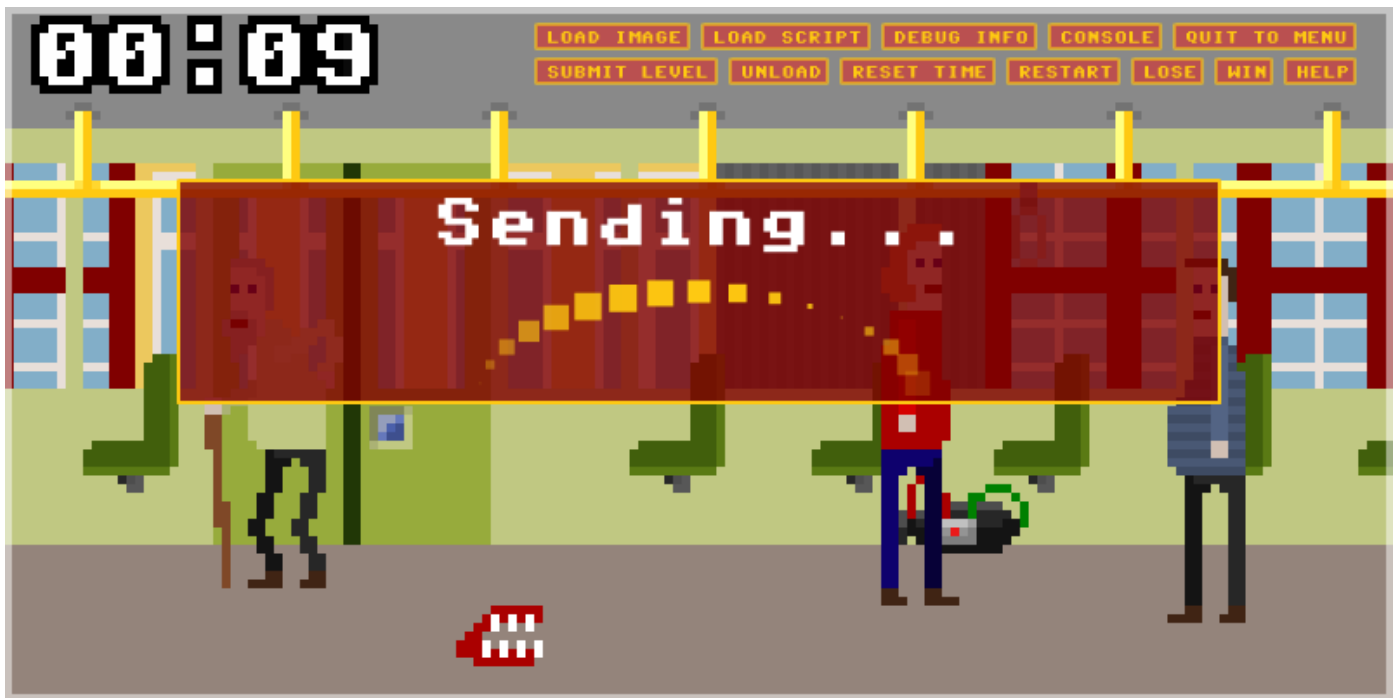
- ❌ **NO LEVEL LOADED** - Level is not running. You need to have level and script that you want to submit loaded.
- ❌ **NO LEVEL** - Level image not loaded
- ❌ **LEVEL HAS ERRORS** - Error was encountered when loading the level. Look at the console to see what it is about.
- ❌ **NO SCRIPT** - Script has not been loaded yet.
- ❌ **SCRIPT HAS ERRORS** - Script has errors, look at the console to check it out.

But since it's pretty simple, I don't think you're going to see these. The submission form is pretty simple too and I already explained it's contents, but for simplicity sake, let's go through that again.



- ❌ **AUTHOR** - Type your name here so that you can be credited for your work!
- ❌ **EMAIL** - Type your e-mail address. I will not be sending any spam. It's just for contact if there is anything wrong with the level that I'd need to discuss before publishing it!
- ❌ **LEVEL TITLE** - Name your level!
- ❌ **NOTES** - If there is anything you'd like to share, express or vent, type it there!

After you filled out the form, tap the... no wait! There is one last thing you need to do. You need to read and accept the legal shit. You can read it if you scroll a bit down, but the form will force you to click the LEGAL SHIT button before you can submit anyways. I suggest you check it out, it's neither long nor obscure. When you're done with that, tap the SUBMIT LEVEL button



If everything goes fine, after few seconds you should see the following message.



This is it! Your level has been sent to my place. It will be published as soon as I get a full sixpack of them. It wasn't that hard, was it? But still, some bad stuff can happen with your submission, and you will be seeing an error message. Let me tell you about these and how to avoid them.

- **CONNECTION BROKE** - You have no internet access or connection with the server was dropped while the level was sending. Make sure your connection is back up and try again!
- **INVALID EMAIL** - The email address you provided is invalid. I really need your email address. I wouldn't want to be blackmailed through the level submission tool.
- **IMAGE MUST BE PNG** - Only PNG format is suitable for the levels, since it's small and supports transparency.

If everything went fine though, just wait until the level is approved and published. I will try to make sure to drop you an email when this happens!

# Appendix A

## Console Errors

Here is a list of errors you can get in the console and a detailed description of their meaning. If you can't see an error and the creator tells you it's there, reload the level or the script to be notified about any errors again.

### SCRIPT ERRORS

- ❌ **TOO FEW LINES!** - The script needs at least two lines, the one saying 'level' and level ID, and the last one saying 'thanks'
- ❌ **BAD HEADER!** - The first line needs at least two parts! The 'level' keyword and the ID
- ❌ **NO 'LEVEL' KEYWORD!** - You didn't type 'level' as the first thing in the file.
- ❌ **BAD ID!** - You are either missing the ID for the header or it's not a number or it's 0 (it can't be 0 by the way)
- ❌ **BAD FOOTER!** - The last line must say 'thanks'.
- ❌ **NO ENDINGS!** - There are no actual endings in this script.
- ❌ **INSUFFICIENT DATA!** - Data in one of the endings is bad. It needs 6 sections separated by '|' (beam) sign
- ❌ **BAD HOLD INFO!** - The hold section must say either 'hold' or 'nohold'
- ❌ **BAD WIN INFO!** - The win section must say either 'win' or 'nowin'
- ❌ **BAD FRAME SPEED!** - Frame speed must be a number between 0 and 1, for instance 0.1
- ❌ **SCRIPT EMPTY!** - The script file is totally empty

### IMAGE ERRORS

- ❌ **LEVEL IMAGE TOO NARROW!** - The image must be at least 4 frames (640 pixels) wide.
- ❌ **IMAGE SIZE MISMATCH!** - the image dimensions must be a multiple of 160 for width, and a multiple of 80 for height.
- ❌ **NO SCROLL INDICATOR!** - The magenta dot that indicates direction and speed of scrolling is missing.
- ❌ **NO PLAYER FOUND!** - McPixel wasn't found in this level. You know he's the main guy there, right?
- ❌ **MORE THAN ONE PLAYER!** - There is only one real McPixel and you shouldn't have two of them on the level.
- ❌ **ITEM ERROR** - Item size can't be 0, items must be at least 1 pixel in size.
- ❌ **NPC ERROR** - Same as above, regarding an NPC.

# Appendix B

## Legal Shit

### §1

I am going to distribute the levels from within the game of McPixel. You're OK with that.

### §2

The level and all the stuff inside is your property. But you let me share it.

### §3

I may be a total ass by refusing to publish your level for no reason. You acknowledge that.

### §4

I can also change stuff in your level before publishing it and still credit you for that. And it's fine with you.

### §5

I can also change your name and level title, but only if it's offensive. I will substitute anything offensive with 'pony' (e.g. *motherponyer*). You like that idea.

### §6

I can delete your level from the server or I can accidentally all the levels. You should keep a copy to yourself!

### §7

Don't sue me bro. That's why i'm writing this.

## Commentary

I'm not good with this kind of stuff, but here it is. This applies to anyone who submits a level. Let me clear some points up as to why I wrote stuff like that. Well, basically I don't want to get sued just because someone is being an ass, but that might make me look like an ass. I guess there's no winning solution in this. Let me account for some stuff that's inside this then.

■ **§3** - I will not be publishing levels that have NSFW stuff inside. Well, at least to some extent. Anyways, there are many things that are gross and not suitable to be shown at all. I will make sure to contact anyone if I have any concerns, but then, I may get the wrong email. Or I may run into a hopeless troll. I don't want to think about it even. Anyways, I will do my best not to be an ass.

■ **§4** - There might be some minor errors in your level that I can fix quicker than I can contact you to fix them and resubmit. Also, if there is some NSFW gross stuff in it, I might want to censor it but still publish the level. It's not like I will draw some awfulness and claim it's someone else who drew that. I just want to feel free to make minor fixes for the level.

■ **§5** - As it says. I will only change it when it's offensive. I won't be changing your name to credit myself or someone else.

■ **§6** - Feel free to let me know if you lost your level data and want a copy, even if it's not published! But I can't promise that I will keep it on the server forever! I will try to keep backups, but still.



# Appendix C

## Graphics Gale Coolness

I am going to tell you about cool features and techniques of Graphics Gale. I've been using it for quite some time now and if you want to create pixel art on Windows, this is the best you can get! Let me now get into details. I will first enumerate features.

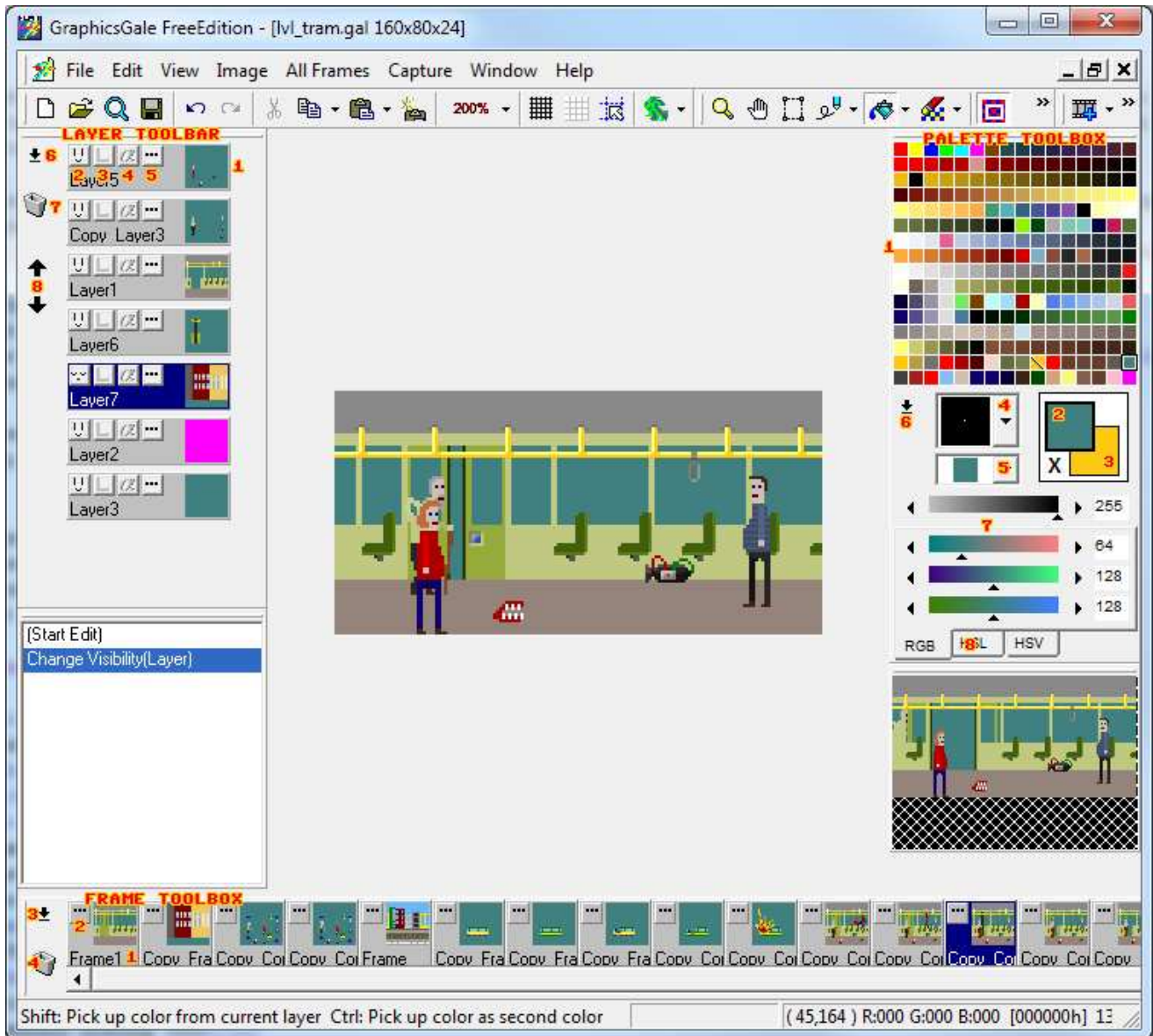
### The weapons



1. **Magnification** - Choose magnification for your image
2. **Grid** - Enable or disable 1 pixel wide grid
3. **Custom grid** - Enable or disable custom grid. You can even customise a list of grid sizes there.
4. **Snap to grid** - Will snap any mouse actions to grid
5. **Onion skin** - This is totally broken or I'm just dumb
6. **Zoom** - Same as 1, but you can click on your image to change the zoom
7. **Mover** - Lets you scroll image with mouse. No idea how this is helpful in anything
8. **Rect selection** - Select a rectangle from your image. After that you can resize, rotate and move it around
9. **Oval selection** - Same as above but with an oval
10. **Lasso** - Freehand selection tool
11. **Magic wand** - Select by colour. You can specify sensitivity too
12. **Colour selection** - Same as magic one... I think
13. **Pen** - Freehand drawing tool
14. **Connect line** - Line drawing. You can draw in two ways. Either by dragging your mouse to draw a single line, or clicking around to draw an interconnected set of lines
15. **Spline curve** - Click around to draw a spline curve
16. **Rectangle** - Draw a rectangle
17. **Filled rectangle** - same
18. **Oval** - Draw an oval (circle)
19. **Filled oval** - same
20. **Flood fill** - You know, the bucket thing. It has a sensitivity setting too!
21. **Color replacer** - It's a really neat tool. Replaces the second colour with first colour. Really helpful in places where flood fill is useless
22. **Draw text** - Let's you draw text on the image. You can set anti aliasing, line spacing and character spacing too
23. **Antialias** - Enables anti aliasing for drawing lines, rotating images and scaling them.
24. **Transparent** - Enables and disables transparency
25. **Paint mode** - There are three modes - paint, clear and opacity. The opacity mode is helpful when using alpha channel









As you can see, this set is clearly missing the eraser tool. This is because instead of erasing, you just draw in a transparent color. Ok, let me move onto describing the toolboxes in more detail.

## The tools



Let's go over all the puzzling buttons scattered across the window. One by one.

### LAYER TOOLBAR

-  **1. Layer picker** - Displays all the layers with thumbnails. You can drag them around to change order
-  **2. Show / Hide** - Toggles layer visibility
-  **3. Lock** - You can lock layer, or lock transparent pixels only. It's a tri-state button
-  **4. Alpha blending** - Toggle alpha blending on and off
-  **5. Properties** - Display layer properties window
-  **6. Menu** - A dropdown menu with some more cool layer options
-  **7. Trash can** - Drag layers there to delete them
-  **8. Move** - Move layers up and down

## FRAME TOOLBAR

1. **Frame timeline** - Displays all frames in a timeline. You can drag these around too
2. **Properties** - Displays the frame properties window
3. **Menu** - Menu with some cool options
4. **Trash can** - Drag frames here to delete them

## PALETTE TOOLBAR

- 1, **Color picker** - The actual palette where you can choose two colours
2. **Primary color** - The colour you are drawing with
3. **Secondary colour** - Almost unused. You can't actually draw with this colour
4. **Brush shape** - You can choose from several brush shapes. Not many, but fair enough
5. **Drawing pattern** - You can choose a drawing pattern along with the brush. So that only some pixels are being filled
6. **Menu** - Few more cool options live there
7. **Colour correction** - Lets you correct your primary colour a bit, or totally
8. **Corrector type** - Choose the type of colour correction tool

Ok, enough descriptions. Let me now point out some cool tricks you can do in the application. I bet you can find them helpful.

In the **Preferences window**, you can alter many settings to your liking. Some of the most useful options there are the ability to choose a **default transparent colour**, and an option to **change every keyboard shortcut** in the application. That's why I didn't give hints about keyboard shortcuts, because you can alter them to your own liking.

There is a useful option called **Make alpha channel** in the **Image menu**. This can create an alpha channel for your image from luminance or selection. You can make alpha channel from another image too!

The next on my list is **Make gradiation** tool from **palette popup menu**. You can set two colours a bit away from each other on the palette, select them both as primary and secondary colour and use this tool to create a smooth gradient of colours between them.

**Right-click color picker** is beyond helpful. This speeds up drawing anything. You can even have a 'palette layer' only for picking colours from it, since you can pick colours from any visible layer (you can of course change that option). Also, you should keep at least a bit of transparent colour visible for picking!

This concludes it. I'm pretty confident you can work something out using Graphics Gale by now!