

MAPMAG

ISSUE 6 - CONCEPT



CONTENTS

The Lobby

Community Interview

MC Buildcon 2017

Big Giant Ideas

New Map - Big Giant Lightshow

Cinematic Tools in Minecraft Mapmaking

Share Command Concepts

Using Generators for your Map

RPG Style Loot

Classic Spawner Concepts

The End... Master Crafter Edition

Image by @Mojang

THE LOBBY

Welcome back to the Minecraft Map Making Scene!

Welcome to the Sixth issue of Map Making Mag. This project started mid-2016 and has successfully grown into a community journal for information about all sorts of things of interest to Minecraft Map Makers. Our past issues have looked at the latest map releases, methods for starting and completing your map making project, and interviews with some very talented Map Makers willing to talk passionately about their projects.

This issue explores the world of the Minecraft Concept creator. Concept creators are often the first people to develop Minecraft features into new mechanics. Their efforts in making game techniques generally available to everyone is much appreciated by the community. We will explore their world in detail this issue.

Where do they get their ideas? How do they make an idea into something that works in a Minecraft world? What happens to the Concept after it is released into the wild? How do people share their discoveries effectively? What does credit mean to these people?

When you see a mainstream media article about the latest amazing Minecraft creation, chances are good that there is a concept creator lurking behind it.

Issue 6 is all about forming deep insights into this lofty realm where innovation happens frequently, with startling results. Special thanks goes to guest editor @CDFDMAN.

If you want to help we are always on the lookout for articles and art. See the side panel for submission guidelines. Until next issue - Happy Map Making!

- Adrian Brightmoore, Editor
Twitter: [@abrightmoore](#)

Check out these Concept Creators:



@SimplySarc



@Dragoz



@HiFolksImAdam



@Sethbling



@JesperTheEnd



@theqmagnet

Submission Guidelines

We are interested in what YOU have to say. Content you make for **Map^{Mag}** can be sent to:
mapmakingmag@gmail.com.

The best letters, articles, art, and other work may be selected for inclusion in **Map^{Mag}** editions or on affiliate websites and other communication channels. Because **Map^{Mag}** is made by the community for the community, **Map^{Mag}** is free for readers and we don't pay you for anything. We ask for permission to include your work in the magazine.

Any content you submit must be your own work, or work that you have the right to submit. By sending us your work you agree that we may edit it for readability or make changes we think are necessary for the magazine. If we decide to include your work you acknowledge that you have granted us the right to publish your work in **Map^{Mag}** and you understand that your work may be quoted or discussed on the internet by anyone in the world without limitation.

All other rights to your work remain with you. You own your work. We are allowed to use it for **Map^{Mag}**. It is that simple.

We will credit you by real name, game name, social media account, or another method that you prefer and that we mutually agree. We will not share your email address without your express permission. If you do not tell us how to credit you for your work then you will not be published in **Map^{Mag}**.

If we refer to you or your work in **Map^{Mag}** you acknowledge that we do so in good will and our intention is not to damage or harm.

DISPUTES

Writing about what you enjoy and hearing from other people with similar interests can be great fun. When people are excited about what they are doing sometimes things can get a little heated in a large community. If you have any concerns over what **Map^{Mag}** is doing or how we are doing it then please contact us describing your concern. This will allow us to understand how we can do better. We can be reached at mapmakingmag@gmail.com.

By reading this magazine you agree that the Contributors, Production Team, and anyone associated with this activity are not liable for any damages to the fullest extent permitted under law. You agree that any dispute arising from this publication is governed by the laws of New South Wales, Australia.

Community Interview

The Map Making community responded to an open survey this issue on the topic of Concepts. This interview has been compiled using their responses. Thank you to everyone who participated.

Where do you get your ideas?

Ideas can come from movies, books, games, other YouTubers, and architecture. Drawing on interests outside Minecraft can prompt interesting thoughts - think of nature, sports, and television shows. Sometimes Minecraft is its own inspiration: Have you looked at all the information in the Wiki entry for ChunkFormat?! Also remember that inspiration can strike anywhere, even in the shower!

How do you make an idea into something that works in a Minecraft world?

Minecraft has no limits, everything can work in Minecraft. It can take hours to make a simple thing though. Command generation websites can speed this up. Spending time experimenting can give you a good understanding of how to work with Minecraft instead of against it. Planning helps. Use pen and paper, or even a Google Doc.

For Command Blocks, tools like CommandStudio are a great help for prototyping. Keeping things modular is always a good idea to keep testing under control.

How do you share your Minecraft discoveries?

Discoveries and concepts can be easily shared on YouTube, chat channels like Discord, the Minecraft Forum, and Twitter. Pastebin is good for sharing interesting commands.

Sometimes a concept can stay unreleased until it can be worked into a Map. If something working can be created, Planet Minecraft and MinecraftMaps.com are good places to host worlds.

Reddit is also an option, though you can be banned for spamming and self-promoting so you have to follow the rules for posting your own content. You can post finished creations to /r/Minecraft with a gfycat link and one - command - install. Discoveries that could be useful to mapmakers (but normal users wouldn't be interested) post to /r/MinecraftCommand.

What happens to your Concept after it is released?

Sometimes it gets played by YouTubers. Impressive concepts can be written about in an article in online media. Feedback helps with the next project.

What does credit mean to you?

Everything. I don't take money for my work so please have at least the decency to write my name.

STEPHEN REID

@IMMERSIVEMIND

Immersive Minds

ICT in Education...

Using technology creatively to enhance learning across the entire curriculum and...

-  Outdoor Education
-  Employability
-  Environmental Science
-  Study Skills
-  Motivation/Aspirations
-  Alcohol Awareness
-  Anti-Bullying/Cyber-Bullying
-  Entrepreneurship
-  Internet Safety
-  Social Media Engagement



Pioneering Games-Based Learning...

Using games and play to enhance and support curriculum learning and life skills development, in children and adults...



Minecraft in Education...

Using Minecraft to support learning across the curriculum...

A global Minecraft server dedicated to training and supporting teachers and parents.



Working with people to develop skills for:

- Work
- Learning
- Life



Communication

Citizenship

Critical Thinking

Numeracy

Analysis

Evaluating

Teamwork

Problem Solving

Creativity

Literacy

Negotiation

Justification

Empathy

Decision Making

Enterprise

Self Confidence

Judgment



MCBuildCon 2017

14th-18th April

MCBuildCon is the annual Minecraft creative Convention hosted online, accessible and enjoyed by all!



Each year we host a variety of events, panels and masterclasses; orientated around the creative side of Minecraft. This year we host over 40 build teams and thousands of players to our biggest BuildCon yet! With a bigger server than ever, and an all new artwork and render showcase, now is the year to come!

The server will lay host to a huge number of team and solo builds alike for you to explore and marvel, as well as a tournament between 24 different trios, battling it out to become the MCBuildCon 2017 Champions. We will be hosting an epic Build Battle between Vaeron and ShapeScape; two of the biggest teams out there.

Masterclasses: events hosted by a master of their field. Whether they be structure, cinematography or renders, you won't want to miss them!

Panels: run by the community to showcase some of the awesome new content, ideas and projects out there in the world of Minecraft!

Showcases: the most epic constructions and creations of the best of the community!

Follow us on Twitter, [@MCBCon](#), for more info on what will be going on, exact server opening times and what events are when!

Big Giant Ideas

How any experience can be created in Minecraft

By: CDFDMAN

It all started with some down time and an empty corner. As Mineathon 2014 was quickly approaching, I was flying around on the server with little to do. I noticed an empty corner in the lobby and began to build a stage. "Why not?", I said to myself. "It'll be a place to hold an opening ceremony or something, who knows."

With the stage complete, there was something missing. Anyone who knows me, knows I'm obsessed with stage lighting; I have been ever since I was a kid. I hopped in a call with my friend SpiderRobotMan and asked him if he had any ideas on controlling color changing beacons. These were a recently added feature in the game and had just been updated to instantly change color. An hour later (and a few villagers) we had beacons that we could program to change colors. I thought, "Wow this is awesome. What else can we do?" Eight hours more and we had the first ever vanilla light show in Minecraft. It featured fireworks, enderdragon beams, and confetti. What started as a time wasting stage build ended up being way larger, and was ultimately recognized by Sarge and Marc of Mojang!



Whether your idea is big to start with or grows from a tiny sapling to a jungle tree, nearly anything from the real world or your imagination can be created in Minecraft. We as creators are no longer confined to the blocks of vanilla or the sweet sounds of C418. With resource packs, command blocks, and custom models; nearly any experience can be created. Granted it won't be a 1 to 1 copy of what you are imagining, but you can still make the general idea. A perfect example of this is the enderdragon beams in my light shows. They aren't what a light beam would look like in the real world but they are close enough to where a player can recognize it as a light beam and enjoy it when they see it as a part of a light show.

The technology of Minecraft doesn't have to hold your ideas back anymore. Dare to dream big giant ideas and most importantly, go out there and make them a reality.

New Map: Big Giant Lightshow

by: CDFDMAN, Sounas, SpiderRobotMan

A year in the making, I present to you the Big Giant Lightshow!

Originally made for Minecraft Realms, this vanilla 1.11 light show is a 20 minute concert experience that is sure to entertain!

The map includes:

- A concert experience unlike any other in Minecraft.
- Never before seen special effects.
- Control the cameras at the concert by switching between inventory slots!
- Amazing music from Big Giant Circles!

This map can be found under the experiences category on Realms or on my YouTube channel /CDFDMAN.



Cinematic Tools in Minecraft Mapmaking: The Cutscene Generator and Transitional Fades

by **The Redstone Scientist (TRS)**

The greatest story-based maps of the last few years all share a few things in common. One of the most obvious commonalities is their cinematic quality. Think about it: Terra Swoop Force, Unsighted Shadow, Exodus Season 2, CDF's Testing Facility series, and many more all drew our attention in unique ways that further elevated them past the realm of "just another map." Cinematic techniques allow us to almost forget that we're playing something in Minecraft. Cutscenes, voice acting, transitions, all of these things which minecraft doesn't naturally have, when added, have a profound effect on the player's experience. It pulls them further into the world of the game and, it can't be denied, produces a "That's pretty dang cool" reaction. I'm going to show you a few tools I've made that can help you get your map counted among the greatest by not only enhancing the "cool" factor, but FAR more importantly, enhancing your story as well.

Let's start with the big kahuna: the Vanilla Cutscene Generator and it's newly released successor "CutscenePRO." The Cutscene Generator has often been hailed as one of the greatest tools to ever enter the mapmaker's toolkit. This tool based in Google Spreadsheets allows you to create cutscenes, cinematic sequences where the player temporarily loses control of their avatar often used for exposition and/or to further the story, with ease. In my opinion, unless your map doesn't call for it, for "big production style maps," cutscenes are an ABSOLUTE MUST. It's a staple of modern triple-A video games and gives you the opportunity to let the player see the world through a perspective they wouldn't normally encounter through regular gameplay. It's also an excellent opportunity to incorporate staples of the cinema industry such as "establishing shots." All these things give your map a professional quality and, if used properly, enhance the understanding and impact of your story.

The Cutscene Generator has two versions: the original and the advanced CutscenePRO. The original has the most basic functionality, allowing you to set a few options, type in up to 500 points, and generate a one command installation that you can install in your world. CutscenePRO is the recently released version which allows you to edit down to the most minute detail your personalized cutscene. Using a unique system of paths, you can even change the time between points. As CutscenePRO is far more complicated, there is a comprehensive manual both written and filmed for you to reference. While these tools are free of charge, donations are appreciated. All I ask is that you give proper credit in your map.



I have a personal interest in ways that we can push the boundaries of mapmaking to strive further towards the cinematic conventions of modern gaming. One of the biggest things I yearned for was the ability to do fade ins and fade outs in Minecraft; both visual fades and auditory fades. After some experimentation, I came up with an easily workable system that you can install in your map. It was originally designed for 1.8 but works just as well in current versions. Now, of course, we have the /stopsound command but it's an abrupt halting of the sound. Unfortunately, the only way we can make music fades work at any time is by using the phenomenon of sound volume based on distance to the origin of sound. This means that if you plan to use music fades in your map, you need to know that from the beginning so the different areas of your map you want to fade between have to be 1000s of blocks apart. But don't write it off just yet! This system also has a resource pack for you to use that enables a function you can install that creates a one second visual fade to black and a one second fade back in. If you want to mix and match these effects, I recommend watching the explanation video linked here:

<https://www.youtube.com/watch?v=ynn6qDWhEug> so you can learn how to easily create your own system that fits your personal needs. To summarize: this system automatically installs two functions you can call with a /setblock command at any time. The first one will fade your screen to black over one second. The second one will fade your screen from black back to normal over one second. The music fades are just a bonus effect that you can learn how to perform in the explanation video. You can see both of these effects demonstrated in this video here:

<https://www.youtube.com/watch?v=yXq1BzC0-0w>

Don't just use these because they are "cool" though. Remember, justify your choices. Enhance your story with an appropriate fade out at a profound moment, get a meaningful close up shot of a door someone is about to walk through, pick and choose your effects so your story really packs a punch.

If you have any questions, feel free to contact me by emailing trs@theredstonescientist.com or tweeting to me @ProfessorTRS.

Cutscene Generator V2.0: <https://theredstonescientist.com/cutsceneprotrailer>

Music Fades and Scene Transitions: <https://www.youtube.com/watch?v=yXq1BzC0-0w>

Music Fades and Scene Transitions Tutorial and Download:
<https://www.youtube.com/watch?v=ynn6qDWhEug>

Guinness World Record
Most Downloaded Minecraft Project



Diversity 2

<https://mods.curse.com/worlds/minecraft/224139-diversity-2>

SHARE COMMAND CONCEPTS

So now you have a beautiful concept, made with command blocks, and you want to share the mechanics? Or you want to help a friend with a command block problem?

CLG is a fully capable online command block editor, but it also comes with the option to share a concept via a short link.

It works like this: You put your commands into the UI, or you import a Smelt project. Configure the type, condition and activation mode. And click on the share button. The website will now compress your commands into a short cryptic code, that, again, gets compressed by Google's url shortening service goo.gl into an only 13 characters short link.

Now you can copy this link and send it to your friend via Discord, Skype, or share it in the description of a video or on Twitter.

Besides that, CLG is a feature-packed command block editor and one command generator. Here is a list of its key features:

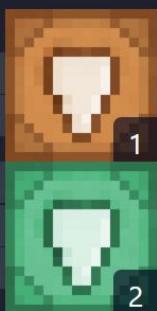
- Intelligent auto-complete
- Project manager
- Find/replace function
- Number and coordinate batch generator
- Custom command aliases
- One command generator for 1.10 and older or 1.11 and newer
- 5 options for row-markers
- Embedded NBT (for command block names and command stats)
- Smelt compatibility
- Built-in selector generator

JannisX11

Tool: <http://server.zofenia.de/clg>

Trailer: <https://youtu.be/WMOVpaQt3kI>

Concept: #summon



/blockdata ~ ~ ~ {auto:0b}

Impulse

Unconditional

Needs Redstone

/scoreboard players tag @e[type=armor_stand,tag=active] remov

Chain

Unconditional

Always Active

Using generators for your map

By Quinten Cabo (IGN: makkie_master)

Using generators for your map.

Since the tellraw command was added to minecraft, its syntax has always been one of the most unforgiving and nasty to write. If you only miss one thing the command will most likely fail.

Luckily generators exist. What are generators you may ask, generators are websites or programs that will just generate the command you need based on the options you choose. These sites have saved countless mapmakers from spending hours just to work on one filled chest setblock, or one complicated summon. A lot of generators exist today. From generating banners and firework to really overpowered golden shovels. There are even generators that make one-commands contraptions for you.

So where do I find these generators! Don't worry I will tell you. Like previously mentioned almost all generators are a site. I will list the ones that are the best in my option.

Mcstacker

It started out as a summoning generator but by now you can use it for almost all command's. Give, setblock, summon. You can use this site as a multitool for basically everything. This is one of the most popular generators, for a good reason I think.

Minecraftjson

This the best minecraft tellraw command generator out there. It has everything you need. Title, books, signs and even the new action bar title. This is an amazing tool.

Mrgarretto.com

This site has useful tools like a command converter from 1.10 to 1.11, an armorstand modeler and best of all the command combiner. This generator this is the end all be all for tools and is basically the best tool on the internet. Constantly improving and updating. It will make a one command from every command you type in. This allows for out of game programming.

In late 2016 Mrgarretto dropped the command combiner pro. This is a full on minecraft compiler.

Command science

This site has a lot of interesting things to choose from. But what stands out the most from this site is its sign generator. There is not a better one around.

Frescoed

All the player heads you will ever need and more!

Minecraft.tools

This site has a lot of cool unique generators. Like a beacon beam color generator, or a nice loot table generator. There is a lot on this site

NovaSkin

If you are a resource pack maker, this is the perfect tool for you! It is essentially an incredibly easy to use but highly advanced resource pack maker. You can make skins with it as well.

There are tons more generators. I thank all the creators for making my life much easier.

By Quinten Cabo
IGN: makkie_master

Using Loot Tables to create...

RPG-style loot

By Cavinator1 ([@ Cavinator1](#))

Introduction

Loot tables are a feature that was added in 1.9. In Minecraft, they control what items mobs drop when killed and what items chests found in dungeons, strongholds, and other structures may contain.

You can edit the vanilla loot tables to make mobs drop different items or chests contain different loot, but the best thing about loot tables, in my opinion, is that you can create your own, and use them in custom maps.

Entity loot tables

In my [Assassin of Steve](#) maps, the gameplay involves defeating mobs and collecting weapons, armor, and potions that they drop so you can use them. As you progress through each mission the loot gets better and stronger.

You can create RPG-style loot from mobs without loot tables, by putting items in their HandItems and ArmorItems slots, however you're limited to the amount of items you can make mobs drop, meaning if you want a zombie that wears full iron armor to be able to drop food, potions, money, or other collectibles, you're gonna need to learn to use loot tables. Loot tables can be created in a text file that needs to be given the file extension .json. In order to edit a .json file you'll need to use a program such as Notepad ++ - the normal Notepad file can't open .json files.

```
1
2   "pools": [
3     {
4       "conditions": [
5         {
6           "condition": "random_chance_with_looting",
7           "chance": 0.16,
8           "looting_multiplier": 0.01
9         }
10      ],
11      "rolls": [
12        {
13          "min": 1,
14          "max": 1
15        }
16      ],
17      "entries": [
18        {
19          "type": "item",
20          "name": "minecraft:cooked_beef",
21          "weight": 1,
22          "functions": [
23            {
24              "function": "set_count",
25              "count": [
26                {
27                  "min": 1,
28                  "max": 2
29                }
30              ]
31            }
32          ]
33        }
34      ]
35    }
36  ]
```

To the left is an example of part of a loot table from Assassin of Steve.

Here you can see a "pool." A pool has a certain number of rolls, and on each roll it chooses an item within the list within. This pool has a condition with a 16% chance of it successfully occurring. There is only one item in this pool's list, which is "cooked_beef," or steak. In the steak entry the function "set_count" sets it so 1 to 2 steak is given. Overall, this means an entity using this loot table has a 16% chance of dropping 1 to 2 steak when you defeat it.

This entity uses many pools to give certain chances of dropping certain items. Further down in this entity's loot table there are pools for dropping equipment. Rather than the pools displaying certain items it links to a different loot table, the loot tables for different types of equipment, as can be seen. The condition means this entity has a 5% chance of drawing from the "army_lance" loot table.

Equipment loot tables

The above loot table shows the Army Lance loot table always drops a stone sword with a function for creating custom nbt. The Name and Lore tags use formatting codes to create a nice-looking tooltip with colors and formatting, as can be seen below. Formatting codes cannot be used in command blocks without third-party software, hence being another advantage for loot tables, as they can. The HideFlags tag removes the usual text on the tooltip that shows enchantments and attributes, as these are shown anyway, and in my own special way, on the custom lore. The AttributeModifier tag makes this weapon deal 6 extra damage when in the player's main hand, so they deal 7 damage total, and also gives the weapon the same attack speed as a normal sword.

Please note that if you use lots of custom attributes for different weapons and armor, make sure the values for their UUIDMost and UUIDLeast tags are all different, otherwise the attributes won't stack if you, say, use a sword at the same time as wearing a helmet that boosts health, or if you use a chestplate that increases movement speed and knockback resistance.

Lastly, below the `set_nbt` function is the `set_damage` function which causes this weapon to drop with between 33% and 66% durability left, similar to mobs in vanilla Minecraft that naturally spawn with armor or weapons.

Army Lance

A monstrously long sword unofficially dubbed a lance

7 Attack Damage
1.6 Attack Speed
Unbreaking
Part of Soldier Equipment

```
25
26     "rolls": {
27         "min": 3,
28         "max": 3
29     },
30     "entries": [
31         {
32             "type": "loot_table",
33             "name": "aos:equipment/army_lance",
34             "weight": 2
35         },
36         {
37             "type": "loot_table",
38             "name": "aos:equipment/soldier_repeater",
39             "weight": 2
40         },
41         {
42             "type": "loot_table",
43             "name": "aos:equipment/bone_shield",
44             "weight": 2
45         },
46         {
47             "type": "loot_table",
48             "name": "aos:equipment/robber_axe",
49             "weight": 2
50         },
51         {
52             "type": "loot_table",
53             "name": "aos:equipment/pilot_helmet",
54             "weight": 2
55         }
56     ],
57 }
```

Using the full loot table shown to the left, you'll get a result that looks something like this:

To the left is the Army Lance's tooltip.

All the weapons, armour and potions use the same sorts of tricks to create custom tooltips.

Chest loot tables

As well as making mobs drop custom items and loot, you can also make chests contain random items by creating loot tables for chests.

In the pool shown to the left there are *three* rolls, and multiple entries in the pool, unlike for the entity loot table shown above. This means that, when the loot table is used it will select three entries from the pool and make the chest contain them in random slots (it may select the same one twice, or even all three times).



Getting entities and chests to use loot tables

To do this, first you need to get into your world's folder. (You can do this within Minecraft by selecting your world, then clicking "Edit," then click "Open Folder.") Inside your world's folder go into the "data" folder, then create a folder called "loot_tables."

Inside the loot_tables folder create a folder and name it anything you want. This is the *domain* of your custom loot tables. Inside your domain folder you can put in your loot tables, or organise it into folders. Suppose you name your domain folder "rpg" and place a loot table inside called "zombie_knight."

The way you get entities to use loot tables is using the DeathLootTable tag. To use your zombie_knight loot table on an entity you would use a command like this:

```
/summon minecraft:zombie ~~~ {DeathLootTable:"rpg:zombie_knight"}
```

If you're working with lots of loot tables it is recommended you sort the loot tables into folders. If you put your zombie_knight loot table inside a folder labelled "mobs" then:

```
/summon minecraft:zombie ~~~ {DeathLootTable:"rpg:mobs/zombie_knight"}
```

Chests use a slightly different tag, plain LootTable rather than DeathLootTable, for example:

```
/setblock ~~~ minecraft:chest 0 replace {LootTable:"rpg:chest/forest_chest"}
```

Conclusion

I hoped you enjoyed reading my article. If you want to learn more about concepts explored in this article, (It would take me many more pages to explain absolutely everything!) read these pages:

[Minecraft Wiki page on Loot Tables.](#)

[Minecraft Wiki page on Formatting Codes.](#)

[Minecraft Forum Loot Table guide by Skylinerw.](#)

I create all my loot tables by hand using Notepad ++ so I cannot exactly say which loot table generator is the best. But here are links to a few:

[Minecraft Tools Loot Table Generator](#)

[Flying Pi Monster's Loot Table Generator](#)

[Mr Pingouin's Loot Table Generator](#)

If you want to learn more about how you're supposed to format loot tables try taking a look at the vanilla loot tables (you can access them using a program like 7-zip, then go into the .jar file for a certain version and into the loot_tables folder), or even the loot tables in a custom map.



Classic Spawner Concepts

There are a selection of concepts that have caused such a high impact on the game that they deserve special mention. These are ideas, and their execution, which resulted in people sitting back and realising that Minecraft is a very complex piece of software worthy of closer attention. We have covered some neat ideas in previous issues, so they won't be in this list. Here we share our personal favourite spawner game mechanic discoveries.



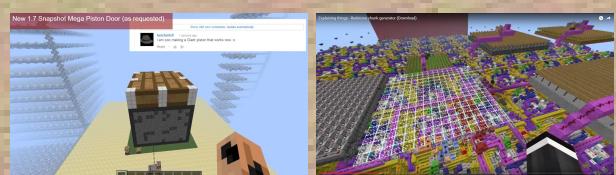
@ZipKrowd learned how to manipulate the world generation from within the game. No MCEdit. No WorldPainter. The method, shown by Panda (in this video: <https://www.youtube.com/watch?v=9iaU1TvIQqM>) was to push blocks into the very edge of the known world using a piston mechanism they called 'cobra' so the game is fooled into believing a Dungeon needs to be populated.

With special care, the team managed to create quadruple spawner dungeons entirely by hand. Impressive stuff!

@Trazlander and @Texelelf collaborated in one of the most useful Map Making discoveries: "Magic Blocks". (https://www.youtube.com/watch?v=Ox5d_wSvDXYk)



As this video shows, structures seem to appear in-game falling layer by layer from the sky and later disappearing without a trace. Blocks change form, like some crazy alchemist is at work. Glass becomes ice. Wood becomes a pumpkin. All before the SETBLOCK command was added to the game, and even before COMMAND BLOCKS existed. The magic of SPAWNERS and FALLING SAND was harnessed by an MCEdit filter made by Texelelf and things were never the same again.



Both **@Twitchnitr0** and **@AsdjkeAndBro** created spawner based contraptions to implement a giant working piston block, and an in-game chunk generator respectively. The versatility of Spawners went on to power Map Making for years!

The End... Master Crafter edition

If all goes well this issue of MapMag is being published around the time Parrots are added to the game. It seems fitting that we consider one of the more prolific concept creators whose appearance on the scene was heralded by the arrival of in-game birds created entirely using vanilla methods (<https://www.youtube.com/watch?v=96OCQmwls4w>)



Onnowhere added a new mob to the game without the support of custom models and without coding extensions through a mod. It was exciting stuff. What made this truly extraordinary was that the creature had unique behaviours and the process was supported with custom crafting recipes and a custom in-game crafting bench.

Onnowhere has been very selective in the projects he releases. For instance he has recreated an entire Minecraft Story Mode episode in vanilla, true to the original game (<https://www.youtube.com/watch?v=On-S5eGi6nE>)

One of the most exciting creations is a Minecraft Artificial Intelligence, named "Albert", as well as the mechanism for a player to 'talk' with it. The AI learns from the player with an expanding knowledge (<https://www.youtube.com/watch?v=IZJUTVh6ge4>).

We also need to consider the influence that mods have had on the game. Sometimes the work of a Minecraft player is so good that Mojang themselves include a version in a future release of the game:

1. Pistons started life as a mod by Hippoplatinus, who is now credited for providing the original code base which the current in-game pistons are based on. Contributions were also made by DiEvAl (<https://redd.it/hvkmo>)
2. Horses started out in the Mo' Creatures mod and were ported into the game with the assistance of the mod author, John Olarte.
3. Slime blocks started as a feature modded into the game by the pervasive and popular @ZipKrowd (<https://www.youtube.com/watch?v=pXBKXJlybg>)

As is frequently the way, slime blocks, in particular, have powered the work of other concept creators. @CubeHamster has created many robots and flying machines, as well as factories capable of generating new devices from raw materials. He has also tried his hand at remaking some classic pop culture references in-game using slime blocks to power them:

<https://www.youtube.com/watch?v=mqyf2i1oH04>

Take some time to get to know the concept creators around you. You may find one day that their amazing ideas become features in your favourite game of Minecraft!

About the Magazine

This project is a community driven and contributed magazine. By publishing we seek to develop the wonderful craft of Minecraft Map Making. All content remains the property of the respective author and is used with permission. All trademarks referenced in this publication remain the property of the respective trademark holder.

Last Issue Errata

Issue 5 - no errors. It turns out it was perfect.

The Map Mag Team

MapMag includes Articles and Art from:

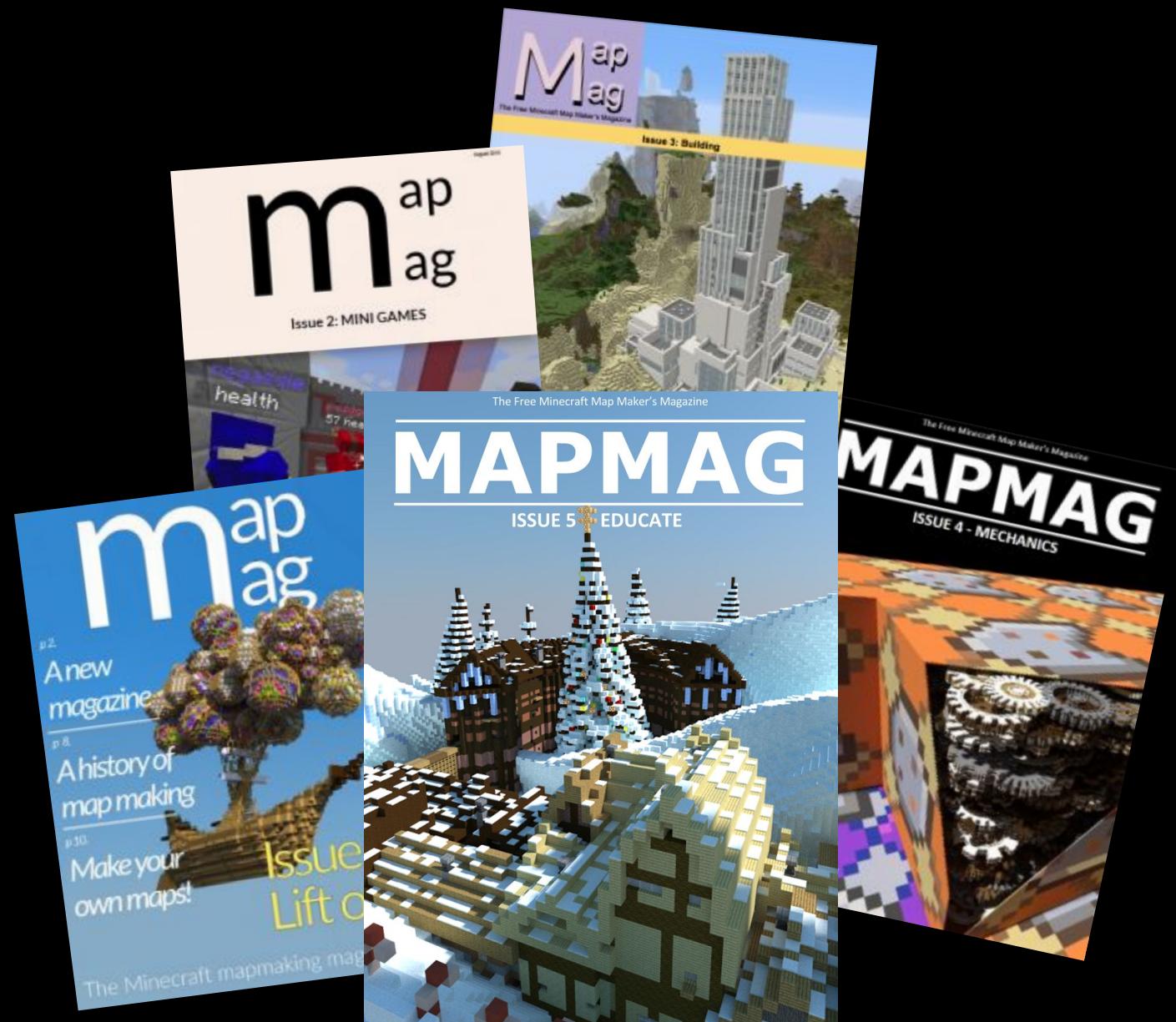
@abrightmoore
@lemoesh
@CDFDMAN
 @_Cavinator1_
 @ImmersiveMinds
 @MCBCon
 @JannisX11
 @ProfessorTRS
 Quinten Cabo
 @TrigonometryBot

@... your name could be here - write an article or provide art for future editions! See submission guidelines in The Lobby.

This publication is a community effort and this issue has been compiled with input from the Minecraft Map Making community. MapMag is supported by donations from: @immersivemind and @cocoamix86

MAPMAG

ISSUES 1-5



<http://www.testfordev.com/MapMag>

@MapMakingMag | MapMakingMag@gmail.com

Images remain Copyright of their respective authors. We use Chunky by Jesper Öqvist and the community (<http://chunky.llbit.se/>) for renders.

We use MCEDIT by @Codewarrior0 and the community (<http://www.mcedit.net>) in the preparation of MapMag