


9장. 마인크래프트 서버 구조

마인크래프트 서버 내부는 어떠한 구조를 가지고 있는지,
네트워크를 어떻게 확장할 수 있는지 알아본다.

플레이어와 UUID 시스템

- UUID란?
 - 오른쪽 그림과 같이 8-4-4-4-12 (30) 형태를 가지는 무작위 문자열
 - UUID 표준에 따라 부여된 값은 고유성을 완벽하게 보장할 순 없지만 실제 사용상에서 중복될 가능성이 거의 없음
- 플레이어의 UUID?
 - Java Edition에서 마인크래프트를 구매하면 플레이어 이름과 함께 UUID가 주어짐
 - 플레이어 이름은 변경할 수 있지만 UUID는 변경되지 않는 고유 값이다.


| Lookup Results: | |
|-----------------|--------------------------------------|
| Username: | solo5star |
| Full UUID: | 2c637bfe-8e80-4e42-a801-b1c703770aef |
| Trimmed UUID: | 2c637bfe8e804e42a801b1c703770aef |



플레이어와 UUID 시스템

- 따라서 플레이어가 자신의 이름을 바꾸더라도 UUID는 고유
- 플레이어 이름으로 차단을 하면 플레이어가 이름을 바꾸어 우회할 수 있다.
- 그러나 UUID를 사용한다면 영구히 차단될 것이다.

| Lookup Results: | |
|-----------------|--------------------------------------|
| Username: | solo5star |
| Full UUID: | 2c637bfe-8e80-4e42-a801-b1c703770aef |
| Trimmed UUID: | 2c637bfe8e804e42a801b1c703770aef |



플레이어와 UUID 시스템

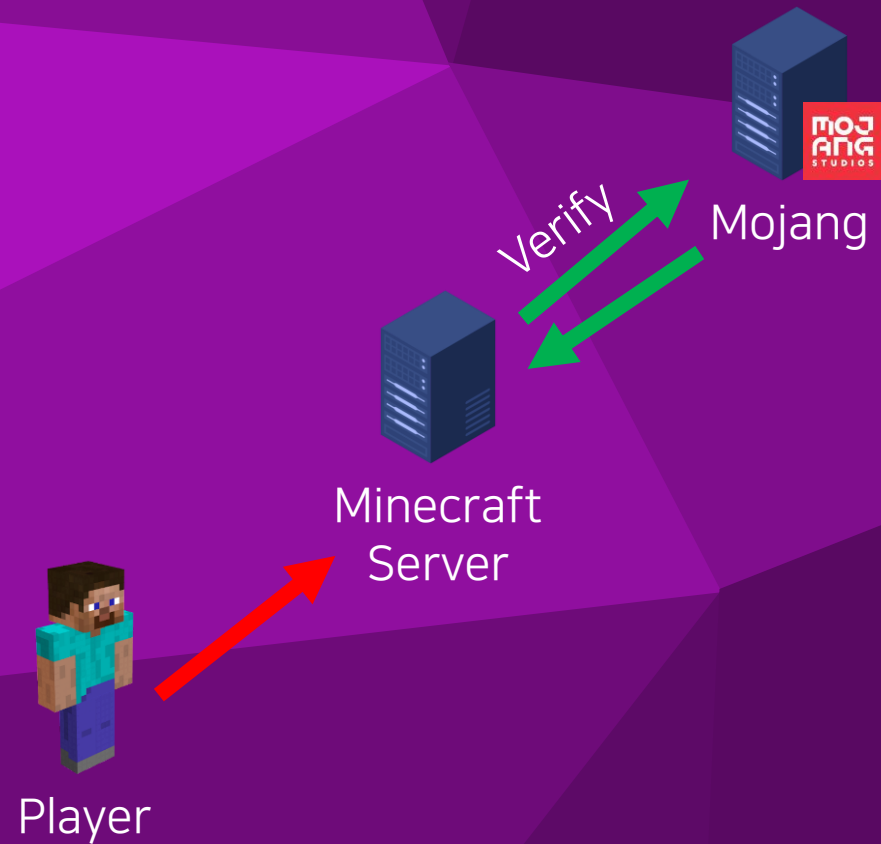
- 플레이어의 UUID는 Mojang에서 발급해주며 API를 통해 조회하거나 특정 사이트에서 조회할 수 있다.
- api.mojang.com/users/profiles/Minecraft/<username>
- <https://mcuuid.net>

A screenshot of a web browser window. The address bar shows the URL 'api.mojang.com/users/profiles/minecraft/solo5star'. The page content displays a JSON response from the API. The response includes a status code of 202 and a JSON object with the player's name 'solo5star' and their UUID '2c637bfe8e804e42a801b1c703770aef'.

```
1 // 20220308235144
2 // https://api.mojang.com/users/profiles/minecraft/solo5star
3
4 {
5   "name": "solo5star",
6   "id": "2c637bfe8e804e42a801b1c703770aef"
7 }
```

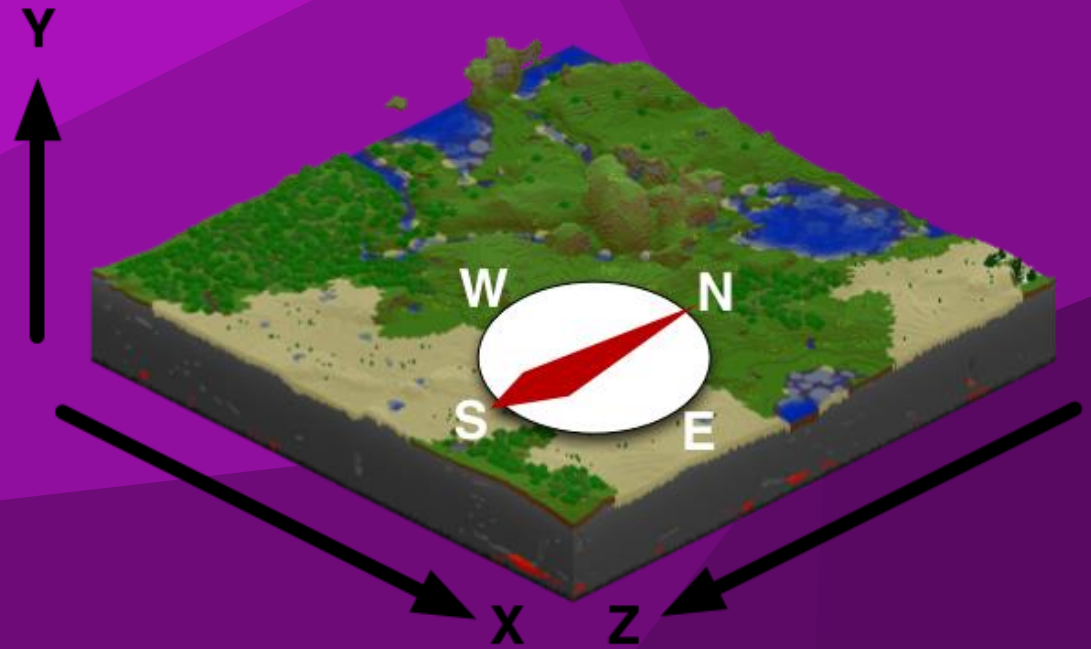
온라인 모드

- 온라인 모드는 접속하는 플레이어를 검증하는 기능이다.
 - Mojang에서 구매한 정품임을 보장받는다.
 - 계정의 주인이 본인인 것을 보장받는다.
 - Mojang에서 발급받은 UUID임을 보장받는다.
- 온라인 모드를 끄면 접속하는 플레이어의 정품 여부와 관계 없이 접속할 수 있게 되며,
 - 계정의 주인이 본인임을 확인할 수 없다.
 - UUID는 랜덤 값으로 사용된다.



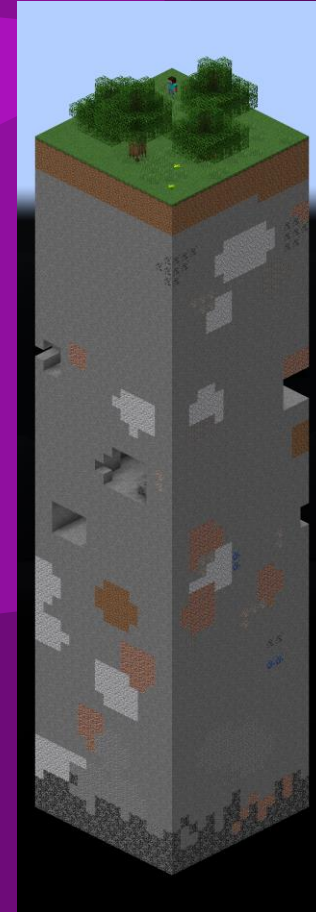
마인크래프트 - 좌표(Coordinates)

- 마인크래프트의 좌표 축은 평면이 x,z 축이며,
- 깊이가 y이다.



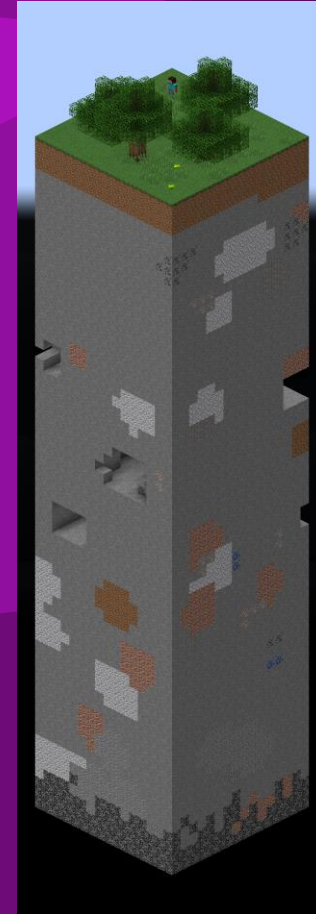
마인크래프트 - 청크(Chunk)

- 마인크래프트의 월드는 무한이다.
- 하지만 월드를 생성하였을 때 처음부터 무한의 월드를 모두 생성하지는 않는다.
- 플레이어 시야에 보이는 부분만 월드를 생성하게 된다.
- 이 때, 생성되는 단위가 청크(Chunk)이다.



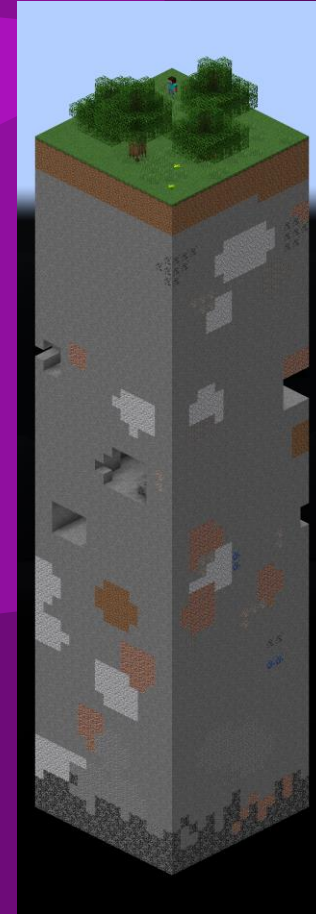
마인크래프트 - 청크(Chunk)

- 월드는 무한이지만 처리를 위한 최소 단위가 필요하다.
- 마인크래프트는 Chunk라는 단위로 처리한다.
- Chunk는 평면 상 16x16 크기의 덩어리이다.
 - y축은 바닥부터 끝까지이다.
- 플레이어가 이동함으로써 월드가 생성될 때,
- 한 블록씩 생성하는 것이 아닌 청크 단위로 생성한다.



마인크래프트 - 청크(Chunk)

- 저장된 월드를 불러올 때도, 월드를 저장할 때도 Chunk 단위로 처리한다.
- 플레이어가 이동하면서 시야에 들어오는 Chunk는 Load된다.
- 플레이어의 시야에서 벗어난 Chunk는 Unload된다.
- Unload된 Chunk에서는 몬스터가 생성되지 않고 작물이 자라거나 물이 흐르는 등의 처리가 수행되지 않는다.



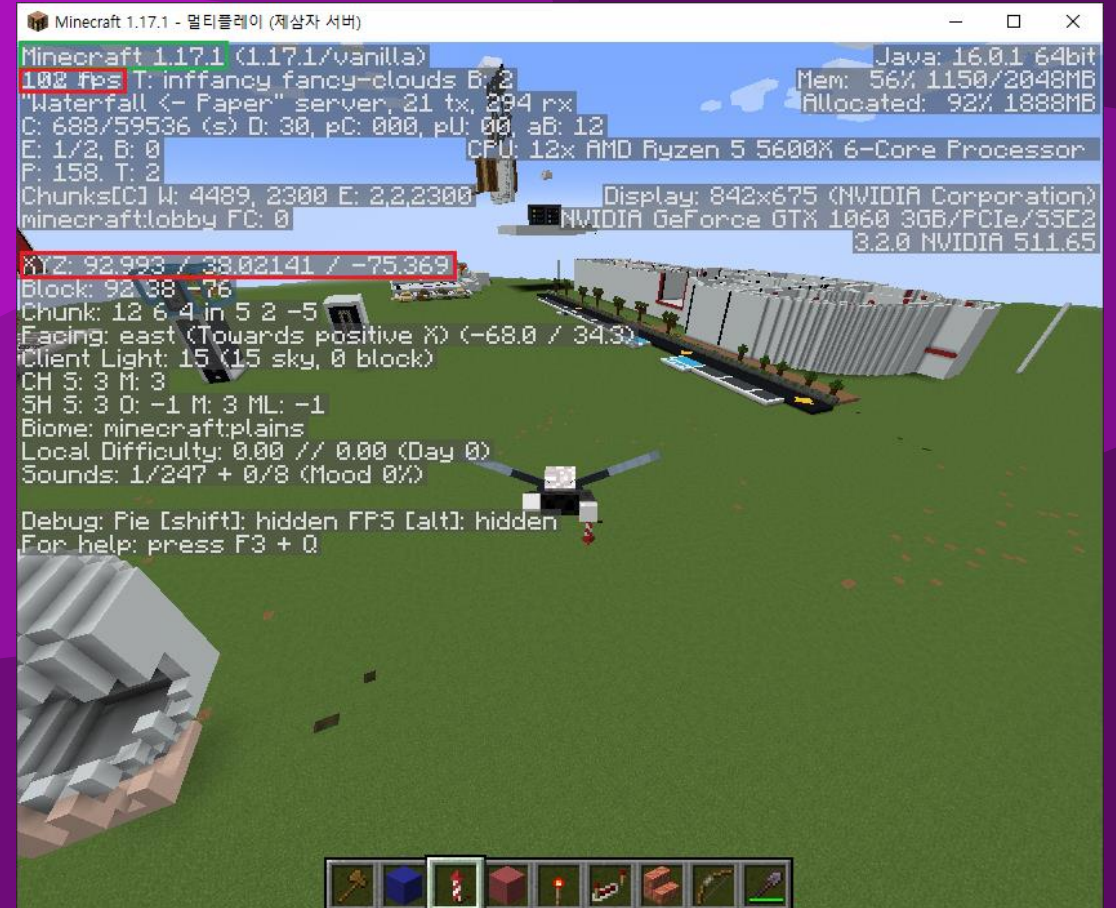
마인크래프트 - 청크(Chunk)

- 서버에서 Chunk는 가능하면 적게 Load된 상태를 유지하는 것이 좋다.
- Load된 Chunk의 수가 많아지면 그만큼 서버의 부하가 증가하기 때문이다.



마인크래프트 - 디버그 화면(Debug Screen)

- 마인크래프트에서는 버전, 좌표, 청크, FPS 등의 디버그 정보를 확인할 수 있는 화면을 제공한다.
- 게임 내에서 F3를 누르면 확인할 수 있다.



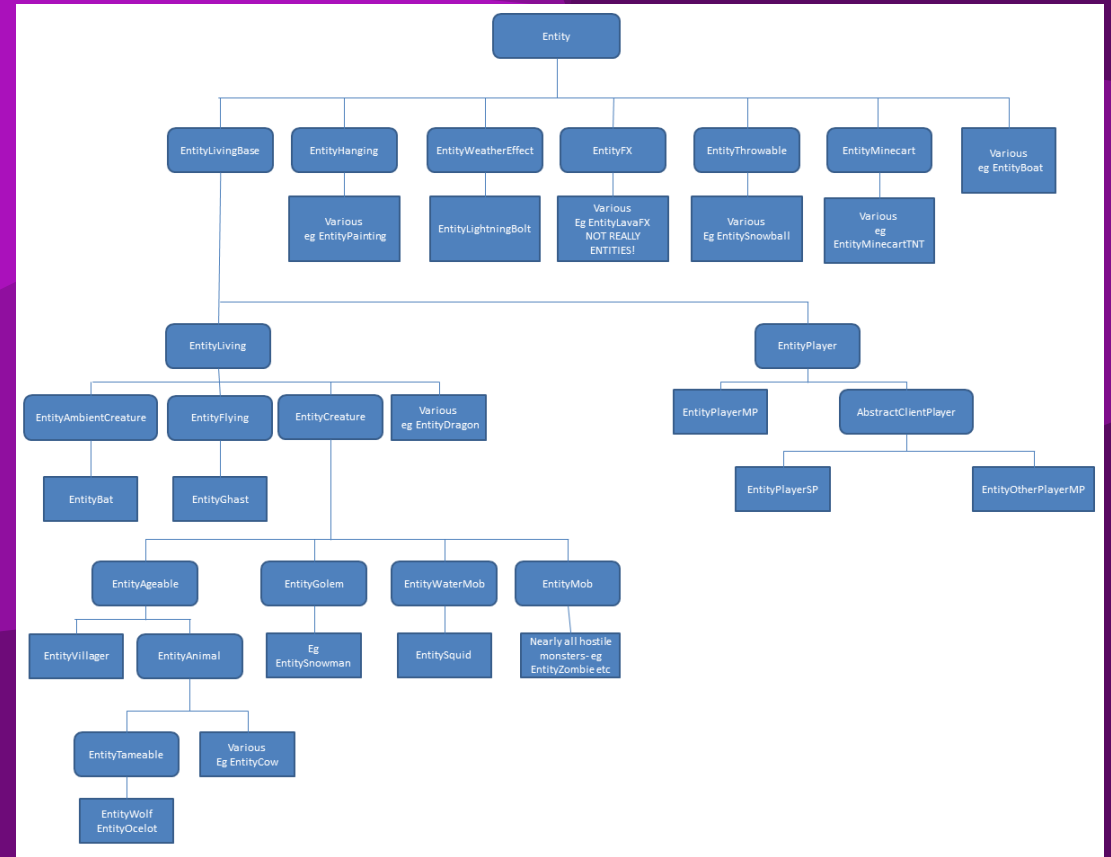
마인크래프트 - 엔티티(Entity)

- 엔티티(Entity)는 마인크래프트 내의 모든 동적인, 또는 움직이는 물체들을 뜻한다.
- 동물, 날라가는 화살, 갑옷 거치대, 플레이어 등의 물체들이 Entity에 해당된다.



마인크래프트 - 엔티티(Entity)

- Entity의 종류는 다양하다.
 - 생물, 날씨 효과(번개), 던진 것(화살, 눈덩이 등), 마인 카트 등 ...
- 오른쪽의 계층 다이어그램을 확인해보자.



구동기 구조 - tick과 TPS

- 구동기가 실행되면 구동기가 종료될 때 까지 while 루프에서 tick을 처리한다.
- tick 처리는 시간의 흐름을 처리하는 것과 같다.
 - 물리 계산, 스케줄링, 패킷 처리 등
 - tick은 1초에 20번 처리된다.
- TPS = Ticks Per Second
 - 1초당 실행되는 tick 횟수
 - 기본적으로 TPS=20 이다.

```
1 public class Server {  
2  
3     // ...  
4  
5     public void run() {  
6         while (true) {  
7             doTick();  
8         }  
9     }  
10  
11     public void doTick() {  
12         // 물리 계산 (플레이어/몹 중력 계산, 충돌 계산 등)  
13         // 스케줄링 (점화된 TNT의 폭발 시간 계산 등)  
14         // 패킷 처리 (플레이어/몹 위치, 시간 흐름, 날씨, 채팅, 월드 전송 등)  
15         // ...  
16     }  
17  
18     // ...  
19 }  
20
```

구동기 구조 - tick과 TPS

- 서버에 부하가 심해지면 TPS가 저하될 수 있다.
 - 1/20초(50ms) 내로 1 tick이 수행되어야 하나, 플레이어가 많거나 서버에 부하가 걸리면 50ms 내로 1 tick을 처리하지 못한다.
 - 처리 시간이 50ms를 넘으면 TPS가 저하된다.
 - 20 -> 18 -> 17... 10 ...
 - TPS=10일 경우 1 tick 처리에 100ms가 걸린다는 뜻
- TPS가 저하되면 유저들은 서버 렉을 체감하게 된다.
 - 다른 플레이어들의 움직임이 뚝뚝 끊겨 보인다.

```
1 public class Server {  
2     // ...  
3  
4     public void run() {  
5         while (true) {  
6             doTick();  
7         }  
8     }  
9  
10  
11 public void doTick() {  
12     // 물리 계산 (플레이어/몹 중력 계산, 충돌 계산 등)  
13     // 스케줄링 (점화된 TNT의 폭발 시간 계산 등)  
14     // 패킷 처리 (플레이어/몹 위치, 시간 흐름, 날씨, 채팅, 월드 전송 등)  
15     // ...  
16 }  
17  
18 // ...  
19 }  
20
```

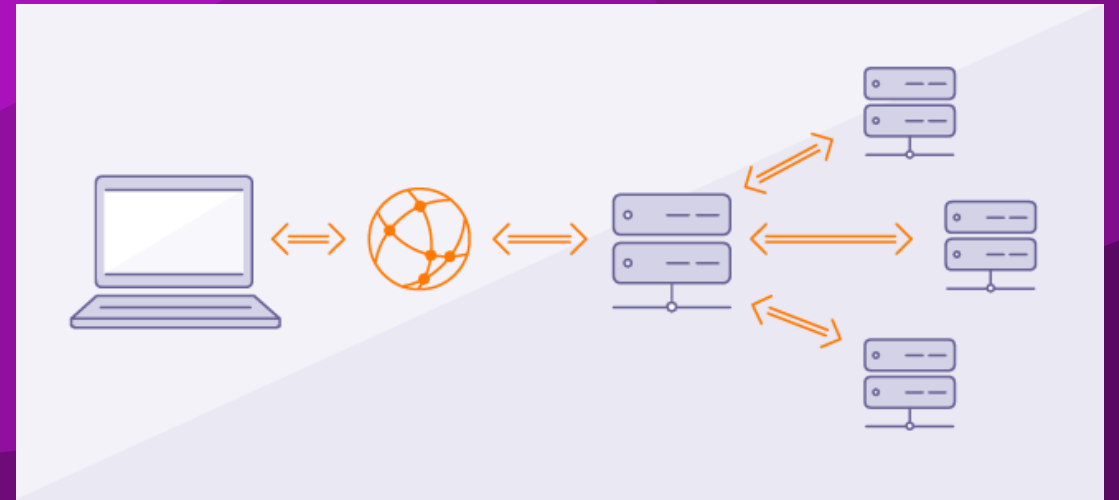
구동기 구조 - 성능 개선

- 플레이어가 많이 접속하거나 Chunk를 많이 로드하면 서버의 부하가 증가한다.
- 50ms 내로 처리되어야 쾌적한 플레이 환경을 유지할 수 있다.
- 하지만 tick 처리는 병렬로 처리되는 부분이 아니기 때문에, CPU 코어의 수를 늘려도 성능 개선이 어렵다.
- CPU의 싱글 코어 클럭 성능을 높이는 것이 바람직하다.

```
1 public class Server {  
2  
3     // ...  
4  
5     public void run() {  
6         while (true) {  
7             doTick();  
8         }  
9     }  
10  
11     public void doTick() {  
12         // 물리 계산 (플레이어/몹 중력 계산, 충돌 계산 등)  
13         // 스케줄링 (점화된 TNT의 폭발 시간 계산 등)  
14         // 패킷 처리 (플레이어/몹 위치, 시간 흐름, 날씨, 채팅, 월드 전송 등)  
15         // ...  
16     }  
17  
18     // ...  
19 }  
20
```


구동기 구조 - 성능 개선

- CPU 싱글 코어 성능을 끌어올리는 데에는 한계가 있다.
- 대신 월드를 분리하고 여러 대의 서버에 분산 처리하여 성능을 개선하는 방법이 있다.



채팅 컬러 시스템

- 구동기에서는 채팅에서 색을 입힐 수 있는 기능을 제공한다.
- 오른쪽의 코드 표를 참고하자.
- 마인크래프트 채팅에서 &aHello &0World! 로 입력하면 Hello는 연두색으로, World!는 검정색으로 표시된다.
- MOTD나 표지판에서도 적용할 수 있다.
 - & 기호가 안된다면 § 기호를 사용해보자.

Minecraft colors:

| | | | |
|----|----|----|----|
| &0 | &1 | &2 | &3 |
| &4 | &5 | &6 | &7 |
| &8 | &9 | &a | &b |
| &c | &d | &e | &f |

| | |
|----------------------|-----------------|
| &k r?u!m | &l Bold |
| &m Strike | &n <u>Uline</u> |
| &o <i>Italic</i> | &r <u>Reset</u> |

```
=====
[12:50:01] [Server thread/INFO]: Welcome!
[12:50:01] [Server thread/INFO]: Online(Cs): 0
[12:50:01] [Server thread/INFO]: Plugins (3): MineCraft
[12:50:01] [Server thread/INFO]: - MineAPI v1.6.1 Craft : Mini-Jeux FR
[12:50:01] [Server thread/INFO]: - AdvancedMotd v Beta 1.1
[12:50:01] [Server thread/INFO]: - AdvancedSignEdit 1.1.0
=====
```

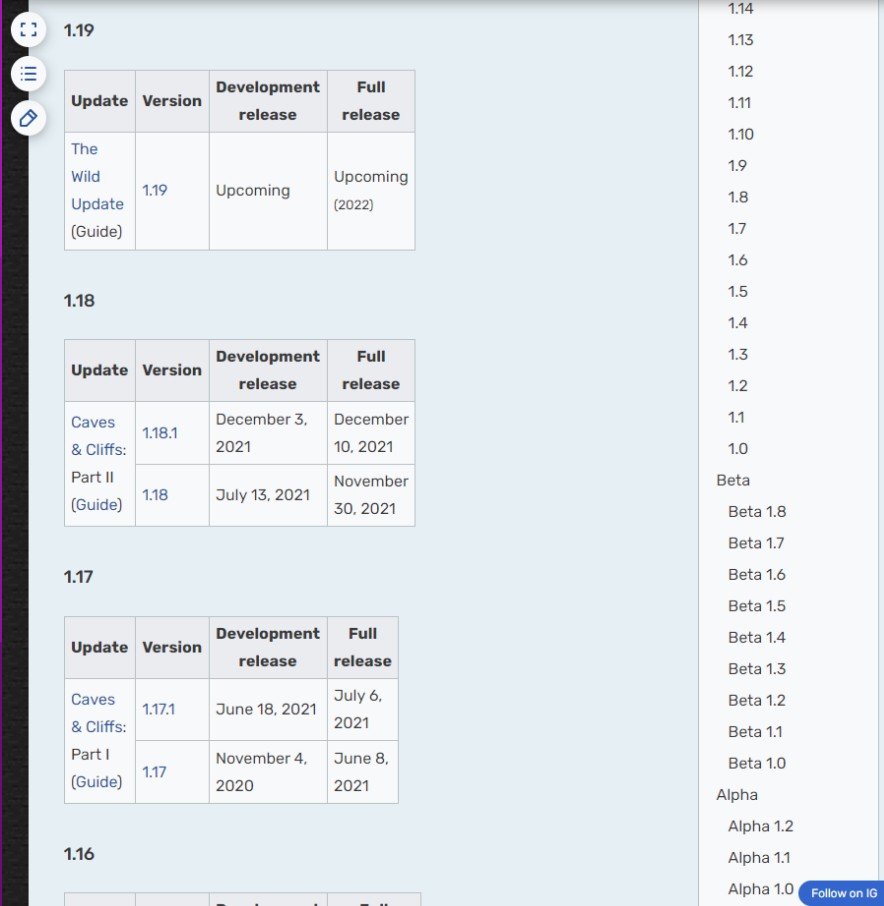
마인크래프트 - 버전(Version)

- 마인크래프트를 개발하는 회사인 Mojang에서는 일정 주기로 버그를 수정하거나 기능을 추가 적용한 업데이트를 배포한다.
- 버전은 x.y.z 형식이다.
 - x는 항상 1이며, 고정 값이다.
 - y는 메이저 업데이트 시 변경된다.
 - 예) Caves & Cliffs 업데이트, Nether 업데이트 등
 - z는 마이너 업데이트 시 변경된다.
 - 버그 픽스 등

[illegible]

마인크래프트 - 버전(Version)

- 버전 이력은 아래의 사이트에서 확인할 수 있다.
 - https://minecraft.fandom.com/wiki/Java_Edition_version_history



| Update | Version | Development release | Full release |
|-------------------------|---------|---------------------|-----------------|
| The Wild Update (Guide) | 1.19 | Upcoming | Upcoming (2022) |

| Update | Version | Development release | Full release |
|---------------------------------|---------|---------------------|-------------------|
| Caves & Cliffs: Part II (Guide) | 1.18.1 | December 3, 2021 | December 10, 2021 |
| | 1.18 | July 13, 2021 | November 30, 2021 |

| Update | Version | Development release | Full release |
|--------------------------------|---------|---------------------|--------------|
| Caves & Cliffs: Part I (Guide) | 1.17.1 | June 18, 2021 | July 6, 2021 |
| | 1.17 | November 4, 2020 | June 8, 2021 |

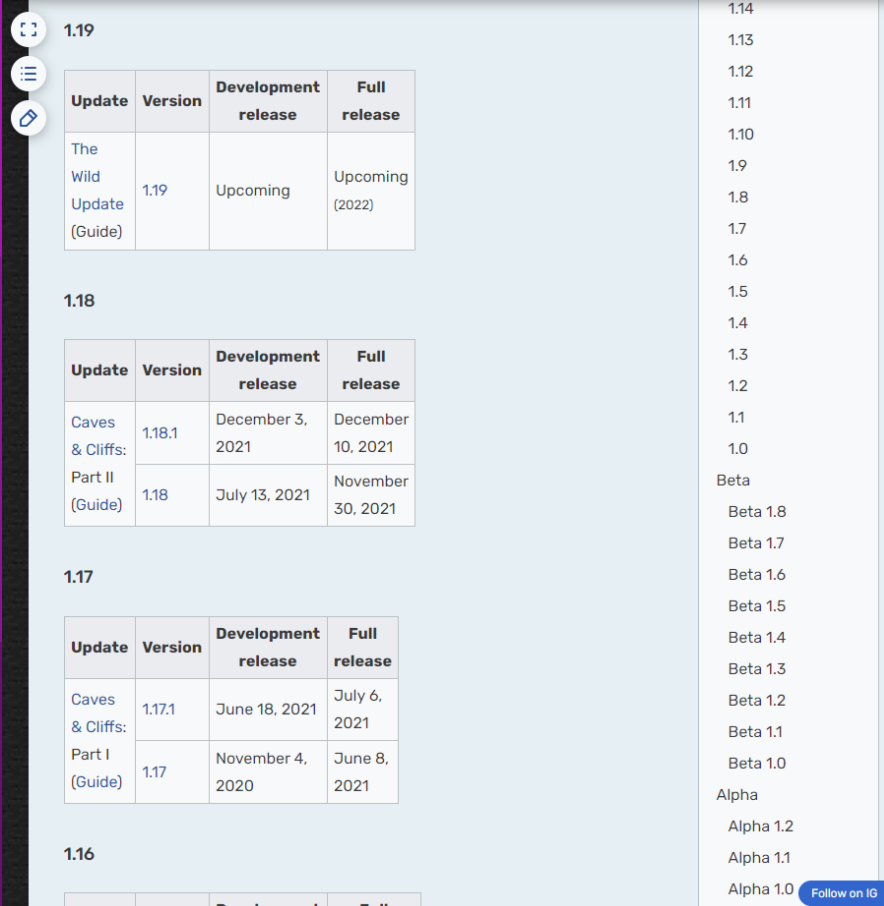
| Update | Version | Development release | Full release |
|--------|---------|---------------------|--------------|
|--------|---------|---------------------|--------------|

1.14
1.13
1.12
1.11
1.10
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1.0
Beta
Beta 1.8
Beta 1.7
Beta 1.6
Beta 1.5
Beta 1.4
Beta 1.3
Beta 1.2
Beta 1.1
Beta 1.0
Alpha
Alpha 1.2
Alpha 1.1
Alpha 1.0

Follow on IG

마인크래프트 - 버전(Version)

- 접속하려는 서버의 버전이 클라이언트의 버전과 일치하지 않으면 접속할 수 없다.
 - 예) 1.18.1서버에 1.17.1은 접속할 수 없다.
- 따라서 유저들이 버전 호환으로 불편을 겪지 않도록 서버의 버전 관리를 해야 한다.



| Update | Version | Development release | Full release |
|-------------------------|---------|---------------------|-----------------|
| The Wild Update (Guide) | 1.19 | Upcoming | Upcoming (2022) |

| Update | Version | Development release | Full release |
|---------------------------------|---------|---------------------|-------------------|
| Caves & Cliffs: Part II (Guide) | 1.18.1 | December 3, 2021 | December 10, 2021 |
| | 1.18 | July 13, 2021 | November 30, 2021 |

| Update | Version | Development release | Full release |
|--------------------------------|---------|---------------------|--------------|
| Caves & Cliffs: Part I (Guide) | 1.17.1 | June 18, 2021 | July 6, 2021 |
| | 1.17 | November 4, 2020 | June 8, 2021 |

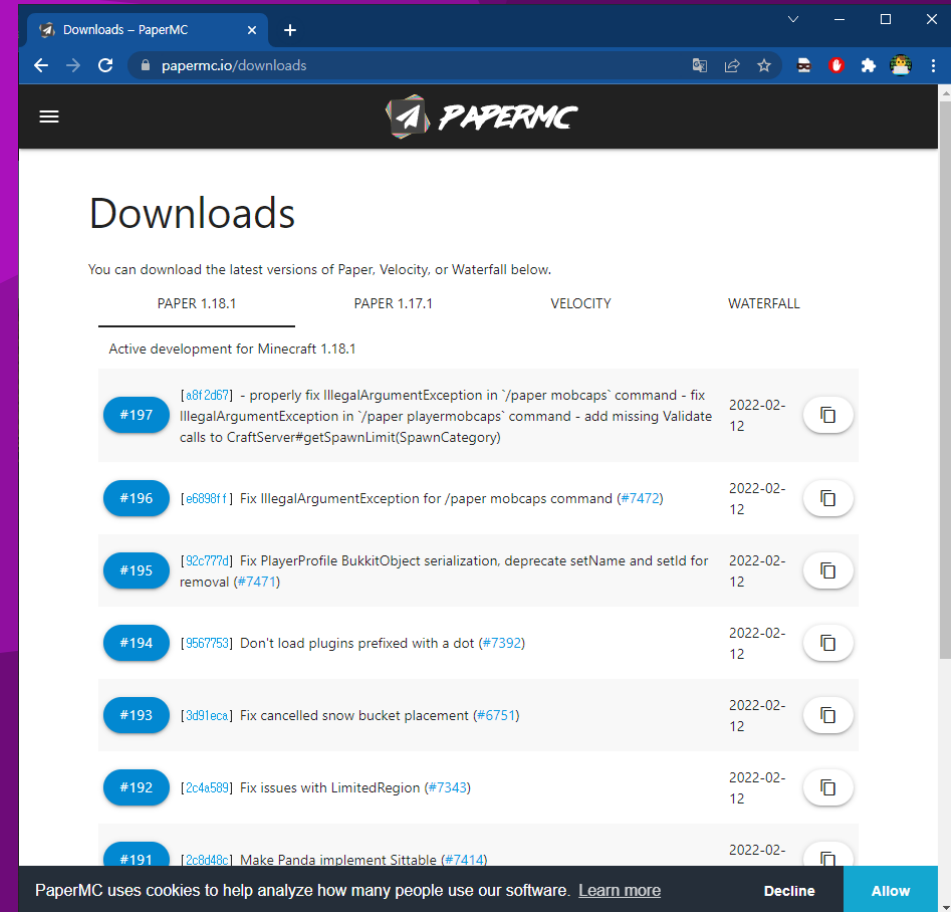
| Update | Version | Development release | Full release |
|--------|---------|---------------------|--------------|
|--------|---------|---------------------|--------------|

1.14
1.13
1.12
1.11
1.10
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1.0
Beta
Beta 1.8
Beta 1.7
Beta 1.6
Beta 1.5
Beta 1.4
Beta 1.3
Beta 1.2
Beta 1.1
Beta 1.0
Alpha
Alpha 1.2
Alpha 1.1
Alpha 1.0

Follow on IG

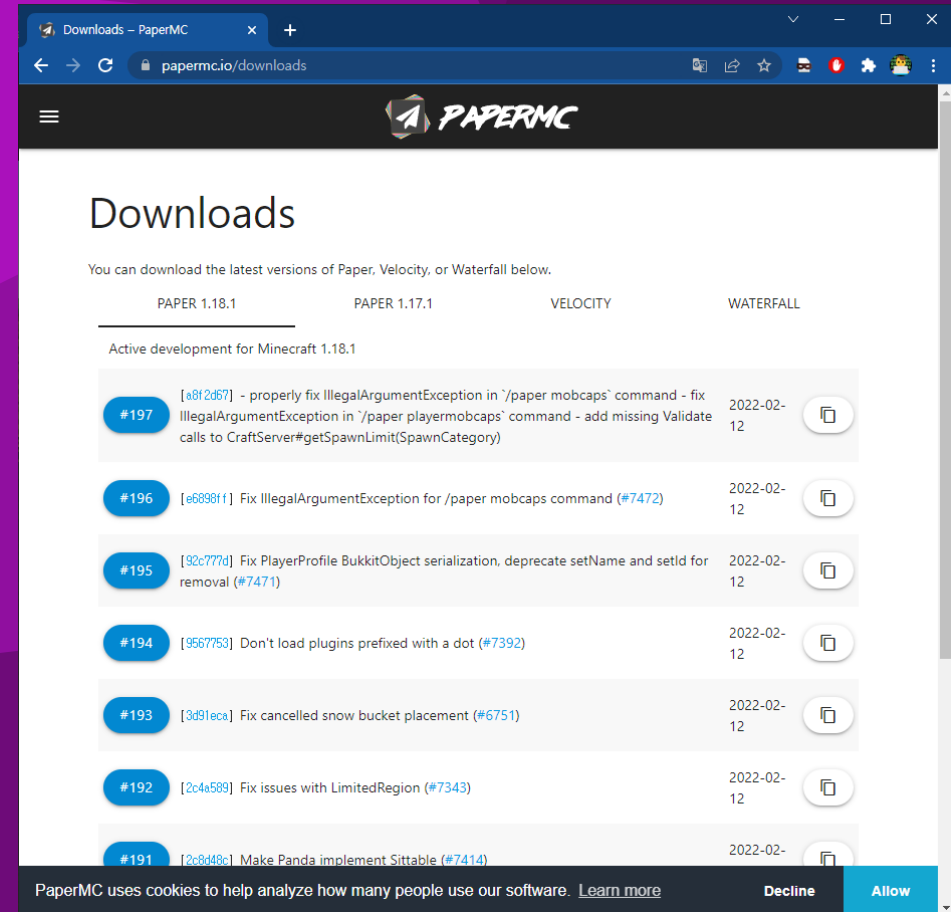
구동기 - 버전(Version)

- Mojang이 공식적으로 제작하고 배포하는 구동기도 있지만,
- 가장 많이 사용되는 Spigot, Paper는 비공식 단체에서 제작하고 배포된다.
 - Spigot Team, Paper Team으로 언급된다.
- 구동기를 제작하는 팀에선 버그 픽스, 기능 추가, 성능 개선 등 구동기를 업데이트하고 배포한다.
 - 이는 구동기 배포 사이트에서 확인할 수 있다.



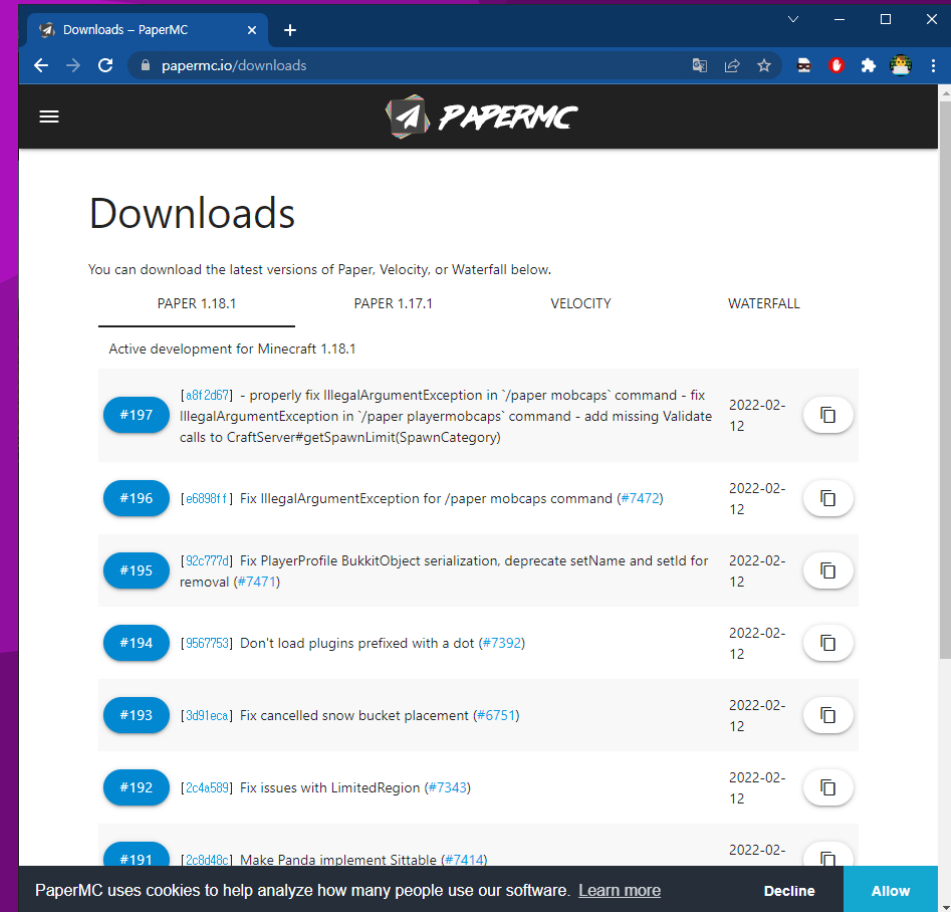
구동기 - 버전(Version)

- 오른쪽은 Paper 구동기 배포 사이트이며, 모든 업데이트에는 빌드 번호가 붙는다.
 - 높은 빌드 번호가 최신 버전이다.
- 운영할 마인크래프트 버전에 맞춰 다운로드 받으면 된다.



구동기 - 버전(Version)

- 빌드 번호가 아닌 마인크래프트 버전을 업그레이드 할 땐 반드시 플러그인 업데이트도 확인하여야 한다.
 - 예) 1.17.1 -> 1.18.1
- 마인크래프트 버전을 올리는 것은 메이저 업데이트에 해당되기 때문에, 플러그인 호환성이 맞지 않을 수도 있기 때문이다.



마인크래프트 - 리소스 팩(Resource Pack)

- 마인크래프트에서는 게임을 확장할 수 있게 해주는 리소스 팩이라는 시스템을 제공해준다.
- 유저는 자신이 만든 텍스처, 모델링, 게임 방식을 리소스 팩으로 압축하여 배포하면, 다른 유저가 이를 다운받아 적용하여 플레이할 수 있다.
- 오른쪽은 현실감 있는 그래픽을 제공하는 BSL 리소스 팩이다.

