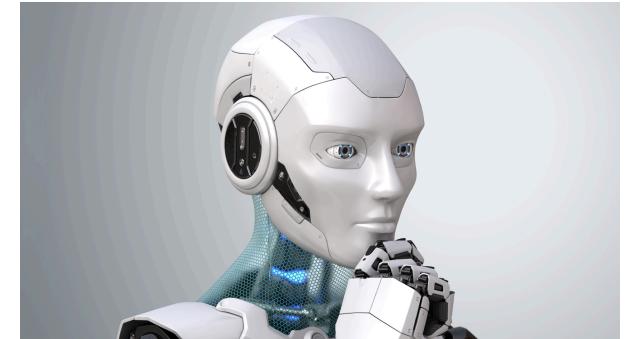


Minecraft bots med Mineflayer



Hva er Mineflayer?

Mineflayer er et kraftig JavaScript-bibliotek som lar deg lage **programmerbare Minecraft bots**.



Hvordan Mineflayer Fungerer

```
const mineflayer = require('mineflayer')

const bot = mineflayer.createBot({
  host: 'localhost', // Server address
  port: 25565, // Server port
  username: 'MyBot', // Bot username
  auth: 'offline' // Auth type
})

bot.on('spawn', () => {
  bot.chat('Hello world!')
})
```

Sette Opp Utviklingsmiljøet

Forutsetninger:

- Node.js (versjon 18 eller nyere)
- NPM (følger med Node.js)
- Kode-editor (f.eks. VS Code)
- Minecraft Java Edition-server
- Alternativt kan du bruke den medfølgende Dockerfilen

Bruke Dockerfilen

- For å bygge Docker-bildet, kjør:

```
docker build -t my-bot .
```

- For å kjøre Docker-containeren, kjør:

```
docker run --network="host" my-bot
```

- Dockerfilen kjører your_code.js

Sentrale Konsepter i Mineflayer

Hendelsesdrevet Programmering

Svar på events i spillet ved hjelp av event listeners:

```
bot.on('chat', (username, message) => {
  if (username === bot.username) return
  bot.chat(`You said: ${message}`)
})
```

Async/Await

De fleste Mineflayer-handlinger returnerer promises:

```
async function goMining() {
  const block = bot.findBlock({ matching: 'diamond_ore' })
  await bot.pathfinder.goto(block.position)
  await bot.dig(block)
}
```

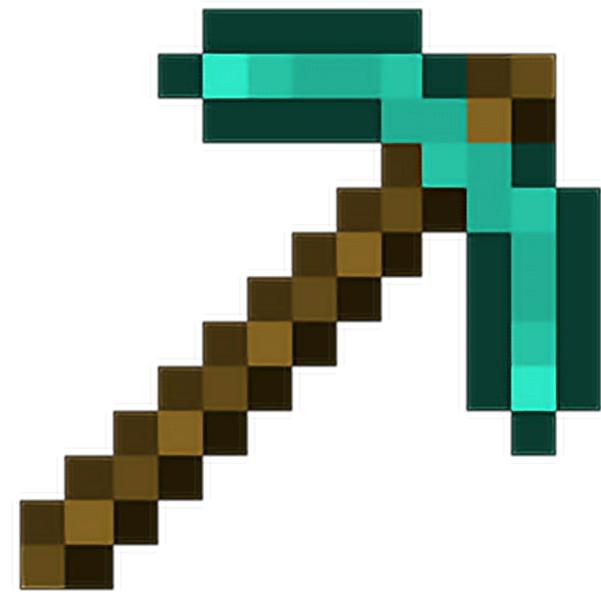
Arbeide med Minecraft-Data

Konvertering mellom block navn og ID-er er en vanlig utfordring:

```
// Initialize minecraft-data with your bot's version
const minecraftData = require('minecraft-data')(bot.version)

// Get block ID from name
function getBlockId(blockName) {
  const block = minecraftData.blocksByName[blockName]
  if (!block) return null
  return block.id
}

// Get item ID from name
function getItemId(itemName) {
  const item = minecraftData.itemsByName[itemName]
  if (!item) return null
  return item.id
}
```



Arbeide med Posisjoner og Vektorer

Bruk alltid Vec3 for posisjonskalkuleringer:

```
const Vec3 = require('vec3').Vec3

// Create a position
const position = new Vec3(0, 64, 0)

// Get position above a block
const blockAbove = position.offset(0, 1, 0)

// Calculate distance
const distance = position.distanceTo(bot.entity.position)

// Get direction vector
const direction = position.minus(bot.entity.position).normalize()
```

Plassere blocks

```
/**  
 * Places a block at a specific position  
 * @param {number} x - X coordinate of reference block  
 * @param {number} y - Y coordinate of reference block  
 * @param {number} z - Z coordinate of reference block  
 * @param {number} face_x - X component of face vector  
 * @param {number} face_y - Y component of face vector  
 * @param {number} face_z - Z component of face vector  
 * @param {object} item - The item to place  
 */  
async function placeBlock (x, y, z, face_x, face_y, face_z, item)
```

- Flatevektoren kan være litt vanskelig å forstå. Den er flaten til referanseblokken som du plasserer blokken på.

Fra Dokumentasjonen

- Flatevektor - en av de seks kardinalretningene, som f.eks. new Vec3(0, 1, 0) for topp-flaten, som indikerer hvilken flate av referanseblokken du plasserer blokken mot.
- Den nye blokken vil bli plassert ved referanseblokk.posisjon.pluss(flatevektor).
- Om du skal plassere en block som ikke er på toppen av en annen block må du teste litt

Eksemelfunksjoner

Jeg har flere eksemelfunksjoner som jeg har hentet fra et annet kurs jeg har holdt. De finnes alle i example.js-filen og er raskt presentert i README.md-filen. De er også importert i your_code.js-filen.



Eksempler på ting boten kan gjøre

- Mange ting du kan gjøre med boten, her er noen eksempler:
 1. **Svare på kommandoer** via chat
 2. **Samle ressurser** automatisk for deg
 3. **Lage gjenstander** når det trengs
 4. **Bygge strukturer** Veldig vanskelig
 5. **Kjempe mot fiender** og beskytte deg
 6. **Utforske verden** og kartlegge den
 7. **Følge deg rundt**
- For å gjøre ting enkelt kan de fleste av disse tingene gjøres ved å kombinere funksjonene jeg har i example.js-filen.

Kjapp Gjennomgang av Eksempel

Takk!

Ressurser:

- Offisiell Dokumentasjon:
<https://github.com/PrismarineJS/mineflayer/blob/master/docs/api.md>
- Navn på Minecraft-Blokker: <https://minecraft-ids.grahamedgecombe.com>
- Eksempler: <https://github.com/PrismarineJS/mineflayer/tree/master/examples>