

```

1  -- Adresářová struktura módu:
2  -- mymod
3  -- textures      -- adresář s texturami (formát png, doporučené
4  --               rozlišení 32x32 či 16x16)
5  -- mymod_node.png
6  -- sounds       -- adresář se zvuky (ogg formát)
7  -- mymod_sound.ogg
8  -- init.lua     -- inicializace módu (vstupní bod - entry-point)
9  -- mod.conf    -- metadata (informace o módu)
10
11
12
13 -- Definice nového objektu
14 -- https://minetest.gitlab.io/minetest/definition-tables/#node-definition
15 minetest.register_node(jmeno, vlastnosti)
16 minetest.register_node(
17 -- jmeno - skládá se z prefixu a jména bločku, prefix by měl být shodný s
18 -- názvem módu, neměl by obsahovat háčky, čárky, mezery ani jiné
19 -- speciální znaky, maximálně " ", např:
20 -- "mymod:node", "mymod:alzak", "mymod:dlouhy_nazev_bez_mezer"
21 "mymod:node",
22 -- vlastnosti - mapování (slovník) který definuje vlastnosti a chování
23 -- bločku:
24 {vlastnost = hodnota,
25  vlastnost2 = hodnota2,
26  vlastnost3 = hodnota3}
27 {
28 -- <<< ZÁKLADNÍ VLASTNOSTI >>>
29 -- description = popis bločku
30 description = "Popis bločku",
31
32 -- is_ground_content = je zemní prvek - může být prvek
33 -- generován pod zemí
34 is_ground_content = true,
35
36 -- groups - skupiny - definuje čím lze objekt vytěžit
37 -- crumbly - hlína, písek - lopatou
38 -- cracky - kamen - kutáčkem
39 -- snappy - listí, drátky, malé rostliny - čimkoliv
40 -- choppy - stromy, dřevo - sekera
41 -- fleshy - živá zvířata, hráči - meč
42 -- exploody - výbušniny - ?
43 -- oddly.breakable.by_hand - louče - čimkoliv
44 groups = {fleshy=2, choppy=1},
45
46 -- drop = upustit - co se z bločku stane po vytěžení, když
47 -- není defiováno, získá hráč tento bloček
48 drop = "default:mese", -- místo tohoto bločku získá hráč mesu
49
50 -- <<< VZHLED >>>
51 -- inventory_image = obrázek v inventáři
52 inventory_image = "mymod_obrazek_do_inventare.png",
53
54 -- tiles = dlaždice - definice jaké obrázky se umístí kam na
55 -- náš bloček
56 tiles = {"mymod_vsechny_strany stejne.png"},
57 tiles = {"mymod_shora.png", "mymod_zdolia.png",
58 "mymod_zprava.png", "mymod_zleva.png",
59 "mymod_zezadu.png", "mymod_zepředu.png"},

```

```

119 -- on_construct, ale je jí předáno více informací (např. kdo
120 -- objekt pokládá, na co se ukazovalo...)
121 -- Pro ukázkou přiřadíme objekt hráči (pokud jej hráč pokládá,
122 -- pokud je pokládá mob či generátor mapy, nic se nestane)
123 after_place_node = function(pozice, kdo, zasobnik, kurzor)
124     if kdo and kdo.is_player() then
125         local meta = minetest.get_meta(pozice)
126         meta:set_string("owner", kdo:get_player_name())
127     end
128 end,
129
130 -- on_punch = při bouchnutí - zavolá se, pokud hráč ukazuje na tento
131 -- položený bloček a zároveň v ruce nemá nic, co by definovalo
132 -- 'on_use' (například nesmí držet jablíčko)
133 -- Pro ukázkou definujeme funkci, která hráči pošle zprávu
134 on_punch = function(pozice, blocek, kdo, kurzor)
135     if kdo.is_player() then
136         minetest.chat_send_player(kdo:get_player_name(), "Nebouchej me")
137     end
138 end,
139 }
140 )
141
142
143 -- Definice nového receptu
144 -- https://minetest.gitlab.io/minetest/definition-tables/#crafting-recipes
145 minetest.register_craft({
146 -- type = typ - druh receptu
147 --   shaped - s tvarem
148 --   shapeless - bez tvaru
149 --   cooking - vaření
150 --   fuel - palivo
151 --   type = "shaped",
152
153
154 -- output = výstup - co bude produktem tohoto receptu, jedná se o text
155 -- kde první část je název bločku, pak mezera a nakonec počet bločků
156 -- na jednotku vstupu (už víte, proč nesmí být mezera v názvu?)
157 output = "mymod:node 32",
158
159 -- recipe = recept - definice receptu. Pro shapeless stačí pole objektů -
160 -- recipe = {"default:dirt", "default:dirt", "default:stone"} -
161 -- pro shaped se jedná o pole polí:
162 -- recipe = {"A1", "A2", "A3"}, {"B1", "B2", "B3"},
163 --           {"C1", "C2", "C3"}
164 -- vyžaduje bločky v "křatičím stolku"
165 --           A1 A2 A3
166 --           B1 B2 B3
167 --           C1 C2 C3
168
169 -- recipe = {
170 --   {"default:dirt", "", ""},
171 --   {"default:dirt", "", ""},
172 --   {"default:stone", "default:stone", "default:stone"}
173 },
174
175
176
177 -- Užitečné funkce

```

```

60 drawtype = druh vykreslení - jak se bude objekt vykreslovat
61 "normal" - základní
62 "allfaces_optional" - zrychlené (pro jednoduché objekty)
63 "glasslike" - sklo
64 "glasslike_framed" - sklo včetně zadní strany
65 "airlike" - neviditelné, bez textur
66 "nodebox" - bloček složený z malých bločků (např. schody)
67 "mesh" - 3d objekt
68 "plantlike" - textury do X
69 "firelike" - podobné jako "plantlike" ale mají efekt na
70 stěnách a stropu
71
72 ...
73 drawtype = "normal"
74
75 -- sunlight_propagates = propouští světlo - je nutné definovat spolu
76 s `paramtype = "light"`.
77 paramtype = "light"
78 sunlight_propagates = true,
79
80 -- <<< FUNKCE VOLANÉ PŘI DRŽENÍ BLOČKU >>>
81 on_use = při použití - funkce, která se zavolá když hráč
82 lehce klikne levým tlačítkem na blok, defaultně se
83 nestane nic
84 on_use = minetest.item_eat(20), -- hráč získá 20 hit-pointů
85
86 -- on_place = při položení - funkce, která se zavolá, když
87 hráč klikne pravým tlačítkem na nějaký bloček a drží
88 u toho náš bloček v ruce, defaultně postaví bloček
89 pomocí `minetest.item_place()`.
90 on_place = minetest.item_place,
91
92 -- on_secondary_use = při druhotném použití - funkce, která se
93 zavolá, když hráč klikne pravým tlačítkem ale neukazuje na
94 jiný bloček (například ukazuje na nebe či jiný objekt),
95 defaultně se nic nestane
96 on_secondary_use = minetest.item_eat(10),
97
98 -- on_drop = při vyhození - funkce, která se zavolá, pokud hráč
99 bloček vyhodí (stiskne `Q`), defaultně vyhodí bloček pomocí
100 `minetest.item_drop`.
101 on_drop = minetest.item_drop,
102
103 -- after_use = po vykutání - funkce, která se zavolá po vytěžení
104 bločku. Zpravidla se používá k opotřebení nástrojů, defaultně
105 dle skupiny bločku
106 after_use = nil,
107
108 -- <<< FUNKCE VOLANÉ KDYŽ MÍŘÍME NA BLOČEK VE SVĚTĚ >>>
109 on_construct = při postavení - zavolá se ihned poté, co je bloček
110 umístěn do světa a je mu předána pouze pozice a nový bloček
111 Zde pro příklad definujeme vlastní funkci, která nastaví
112 popis bločku, který se ukáže po ukázání na bloček
113 on_construct = function(pozice, blocek)
114     local meta = minetest.get_meta(pozice)
115     meta:set_string("infotext", "Tohle je blocek")
116 end,
117
118 -- after_place_node = po umístění bločku - volá se obdobně jako

```

```

178 -- Poslat uživateli textovou zprávu
179 -- minetest.chat_send_player(jmeno, zprava)
180 minetest.chat_send_player(clicker:get_player_name(), "Alza vyprodej!")
181
182 -- Přehraj zvuk uživateli
183 -- minetest.sound_play(nazev_souboru_bez_pripomy, definice_zvuku)
184 minetest.sound_play("ldoktor_alza_vyprodej", {
185     -- to_player = kterému uživateli
186     to_player = clicker:get_player_name(),
187     gain = 1.0,
188     -- max_hear_distance = maximální slyšitelná vzdálenost
189     max_hear_distance = 32,
190     -- loop - přehrávat opakovaně
191     loop = false})
192
193 -- Najdi bloček v okolí
194 -- minetest.find_node_near(pozice, max_vzdálenost, druh_bločku)
195 minetest.find_node_near(pozice, 50, { "default:mese" })
196
197 -- Umísti/mahrad bloček na pozici
198 -- minetest.set_node(pozice, druh_bločku)
199 minetest.set_node(pozice, { name = "default:mese" })
200
201 -- Nastav metadata bločku (např. popis bločku)
202 -- nejprve musíme získat metadata bločku na známé pozici
203 local meta = minetest.get_meta(pozice)
204 -- nyní nastavíme "infotext" metadata
205 meta:set_string("infotext", "Jsem tak otravněj")
206

```