

# Methoden "ChallengeBox" und "TeamMatrix" für Herausforderungen und Gruppenzuordnungen



Verteile Lernende zufällig auf Gruppen oder zeige zufällig ausgewählte Herausforderungen mit einem mit LUA programmierten Gerät an.

## Methode

Nutze programmierbare Geräte, um Lernende zufällig in Gruppen einzuteilen oder bestimmte Herausforderungen zu verteilen. So kann die ChallengeBox schnell angepasst werden, um zum Beispiel eines der 17 Nachhaltigkeitsziele zufällig auszuwählen oder das Verteilen mit per [wesh-Modifikation](#) miniaturisierter Wichtelgeschenke zu erleichtern. Die TeamMatrix teilt Lernende ganz einfach in Gruppen oder Bereiche ein, wenn sich die Avatare vor eine Kamera stellen.

Typ	Gruppeneinteilung, zufällige Herausforderungen
Vorbereitung	15 - 20 min
geeignet für	Lernende mit ersten Erfahrungen mit Luanti oder begleitete Lernende

# Vorgehensweise

## ChallengeBox (Zufällige Herausforderungen)



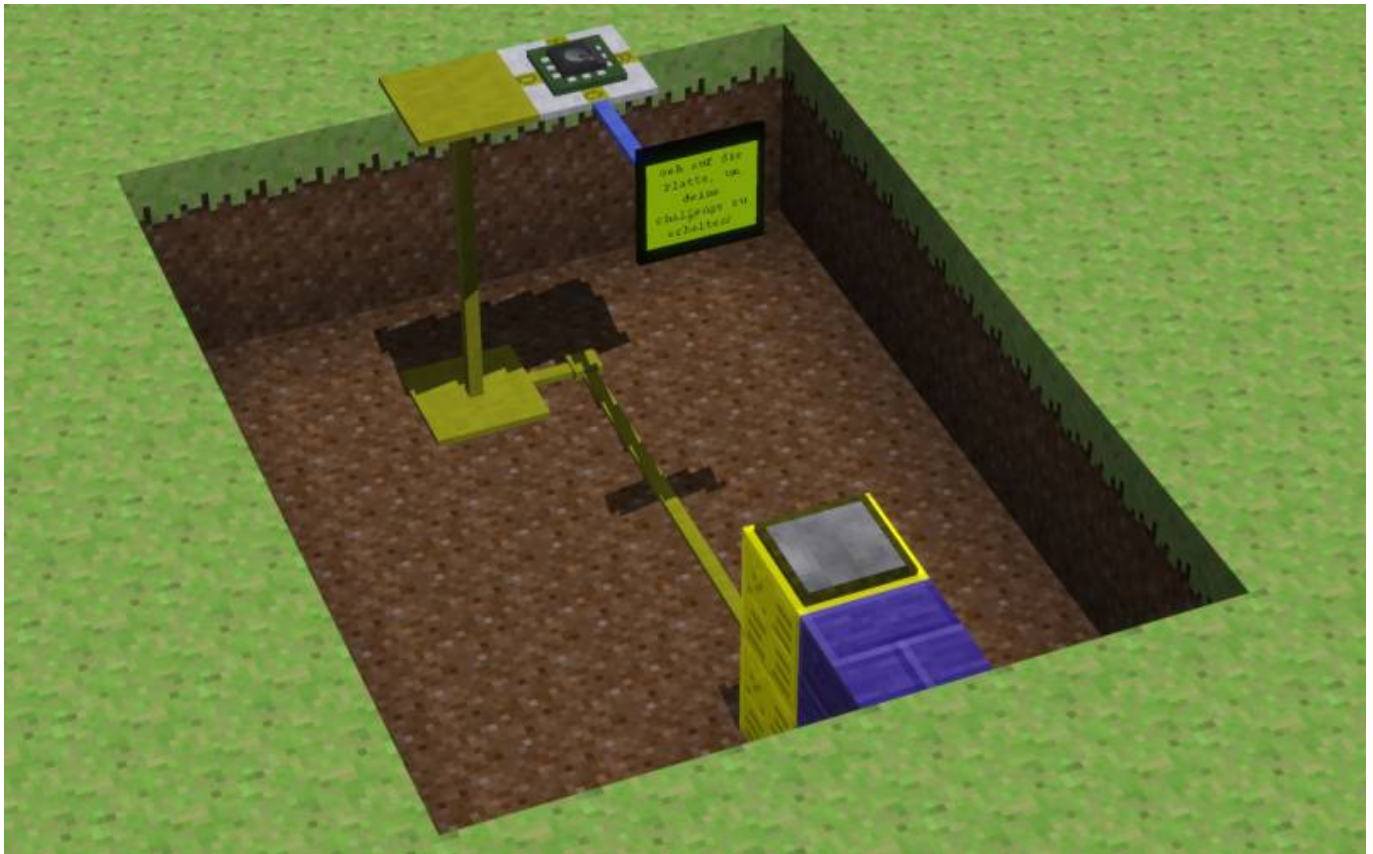
Baue ein Gerät mit einem LUA-Controller, das aus einer Liste mit zufälligen Herausforderungen eine auswählt und immer dann anzeigt, wenn jemand sich auf eine Platte stellt oder einen Knopf drückt. Du kannst die ChallengeBox zum Beispiel verwenden, um eines der [17 Nachhaltigkeitsziele](#) auszuwählen, Themen für ein neues Bauwerk anzuzeigen oder beim Wichteln Anweisungen zu geben, in welche Richtung das Geschenk weitergegeben oder getauscht werden sollte, wenn Wichteln mit Tauschregeln gespielt wird. Deiner Kreativität sind hier keine Grenzen gesetzt.

Für die ChallengeBox benötigst du

- einen LUA-Controller
- [Mesecon](#)-Kabel, flach und vertikal
- (optional) einen Meseblock
- Digiline-Kabel
- einen Bildschirm
- eine Druckplatte oder einen Taster

Verbinde die Gegenstände wie im Bildschirmfoto aufgezeigt miteinander. Unterhalb des Mesecon-Kabels kannst du einen Meseblock platzieren, der ebenfalls ein Signal weitergibt, aber einen ganzen Block einnimmt. Stelle den Bildschirm per Rechtsklick auf einen festen Kanal ein, damit er die Ausgabe anzeigen kann. In unserem Codebeispiel haben wir dem Bildschirm den Kanal `screen` gegeben. Füge anschließend den unten gezeigten Programmcode per Rechtsklick in den LUA-Controller ein und klicke auf Execute.

[Aufbauanleitung als Bild](#)



### Programmcode für LUA-Controller

```

if event.type == "program" then
    -- Füge hier die Herausforderungen oder Anweisungen ein
    mem.herausforderungen = {"Ziel 1", "Ziel 2", "Ziel 3"}
    -- Füge hier den Text ein, der angezeigt wird, wenn kein Ziel angezeigt
    wird
    mem.hint = "Geh auf die Platte, um deine Challenge zu erhalten!"
    -- Gib hier die Zeit in Sekunden ein, nach der der Hinweistext angezeigt
    werden soll
    mem.cooldown = 2

    interrupt(0.2, "idle")
end

local function clearScreen()
    for i=1,6,1 do
        digiline_send("screen", " ")
    end
end

if event.type == "interrupt" and event.iid == "idle" then
    clearScreen()
    digiline_send("screen", mem.hint)
end

if event.type == "on" then
    clearScreen()

```

```
digiline_send("screen", "hello")
digiline_send("screen",
mem.herausforderungen[math.random(#mem.herausforderungen)])
interrupt(mem.cooldown, "idle")
end
```

## TeamMatrix (Gruppen- oder Herausforderungszuordnungen)



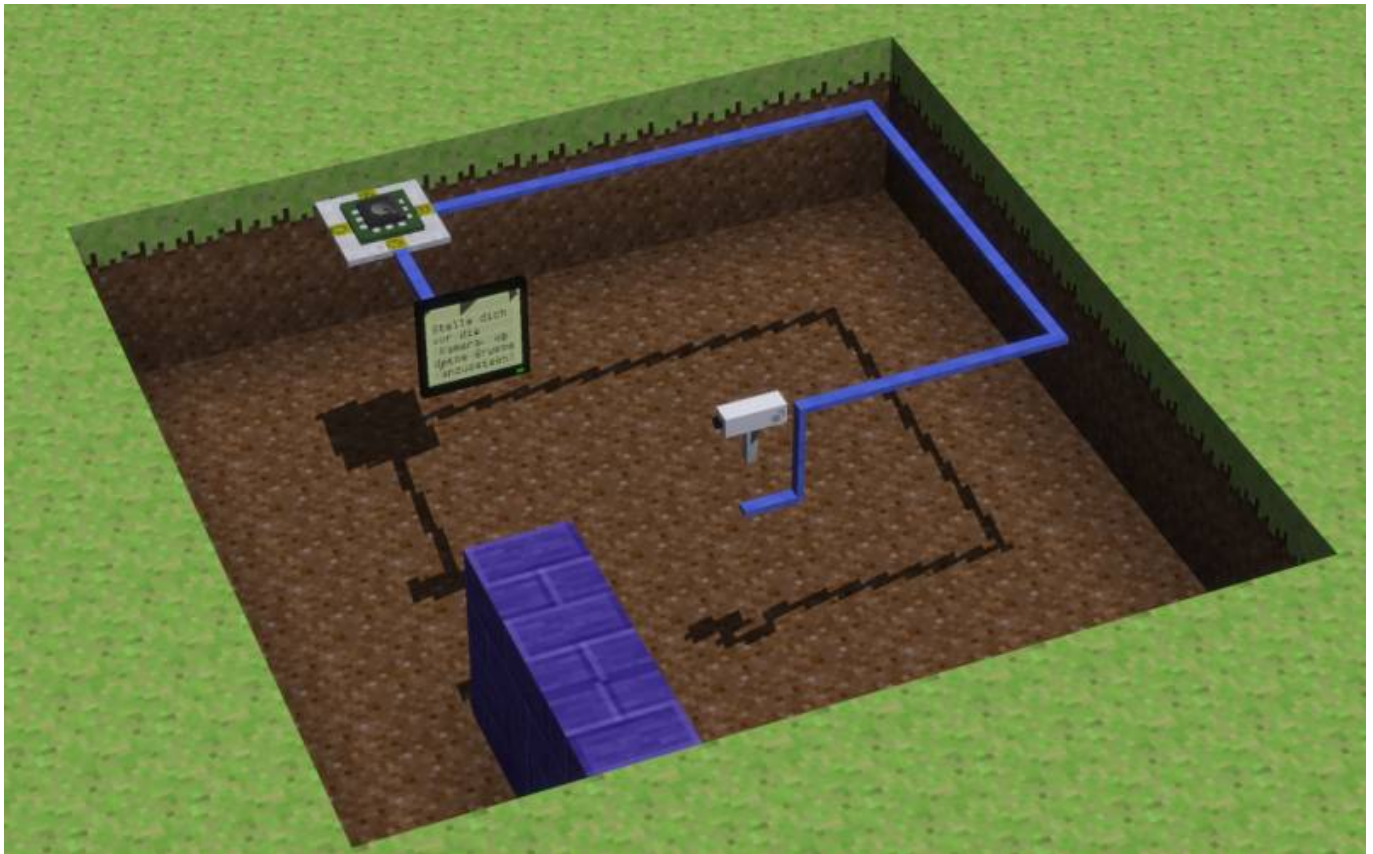
Baue ein Gerät mit einem LUA-Controller und einer Digilines-Kamera, das Lernende und Lernbegleitende zufällig einer Gruppe zuordnet. So können Lernende sich zufällig einem Team zuordnen, um kollaborativ zu planen, zu bauen oder miteinander in den Wettstreit zu treten.

Für die TeamMatrix benötigst du

- einen LUA-Controller
- Digiline-Kabel
- einen Bildschirm
- eine Digiline-Kamera

Verbinde die Gegenstände wie im Bildschirmfoto aufgezeigt miteinander. Stelle den Bildschirm per Rechtsklick auf einen festen Kanal ein, damit er die Ausgabe anzeigen kann und konfiguriere die Kamera per Rechtsklick ebenfalls, damit sie den korrekten Bereich filmt. In unserem Codebeispiel haben wir dem Bildschirm den Kanal `screen` und der Kamera den Kanal `camera` gegeben. Füge anschließend den unten gezeigten Programmcode per Rechtsklick in den LUA-Controller ein und klicke auf Execute. Die TeamMatrix kann auf die Anzahl der Lernenden eingestellt werden, um die Lernenden gleichmäßig zu verteilen. Das Gerät merkt sich außerdem, wer welcher Gruppe zugeordnet wurde und zeigt diese Information erneut an, wenn jemand vor der Kamera steht. Außerdem kannst du die Gruppennamen selbst einstellen. Verändere dazu die Zeilen `mem.lernenden_count` und `mem.groupnames` wie im Beispiel gezeigt. Im Codebeispiel ist außerdem beschrieben, wie du die Hinweis- und Fehlertexte ändern kannst.

[Aufbauanleitung als Bild](#)



### Programmcode für LUA-Controller

```

if event.type == "program" then

    -- Stelle hier die Anzahl der Lernenden ein, die verteilt werden sollen.
    mem.lernenden_count = 4
    -- Stelle hier die Gruppennamen ein.
    -- Achte darauf, dass die Namen in Anführungszeichen sind und trenne sie
    mit einem Komma.
    mem.groupnames = {"a", "b"}
    -- Stelle hier den Hinweis ein, der erscheinen soll, wenn niemand vor der
    Kamera steht.
    -- Mit \n kannst du eine neue Zeile auf dem Bildschirm erzwingen
    mem.hint = "Stelle dich vor die\n Kamera, um deine Gruppe anzuzeigen!"
    -- Stelle hier den Hinweis ein, der erscheinen soll, wenn alle Gruppen
    voll sind.
    mem.hint_vollegruppen = "Alle Gruppen sind voll!"
    -- Stelle hier den Hinweis ein, der erscheinen soll, wenn mehrere Avatare
    im Kamerabereich stehen
    mem.hint_eineperson = "Bitte nur eine Person vor der Kamera!"
    -- Stelle hier die Zeit in Sekunden ein, die die Gruppenzugehörigkeit noch
    angezeigt werden soll,
    -- nachdem ein Avatar den Kamerabereich verlässt
    mem.cooldown = 2

    mem.groups = {}
    mem.seen_lernende = {}

```

```
for _, groupname in ipairs(mem.groupnames) do
    mem.groups[groupname] = {}
end
interrupt(0.2, "idle")
end

local function clearScreen()
    for i=1,6,1 do
        digiline_send("screen", " ")
    end
end

if event.type == "interrupt" and event.iid == "idle" then
    clearScreen()
    digiline_send("screen", mem.hint)
end

local function groupHasCapacity()
    for _, groupname in ipairs(mem.groupnames) do
        if (#mem.groups[groupname] or 0) < mem.lernenden_count /
#mem.groupnames then
            return true
        end
    end
    return false
end

local function assignPlayer(player)
    if not groupHasCapacity() then
        digiline_send("screen", mem.hint_vollegruppen)
        return
    end

    local assigned = false
    while not assigned do
        local selected_group = mem.groupnames[math.random(#mem.groupnames)]

        if (#mem.groups[selected_group] or 0) < mem.lernenden_count /
#mem.groupnames then
            table.insert(mem.groups[selected_group], player)
            assigned = true
        end
    end
    table.insert(mem.seen_lernende, player)
end

if event.type == "digiline" and event.channel == "camera" then
    clearScreen()
    if #event.msg == 1 then
        local player = event.msg[1]
```

```

-- Überprüfen, ob Avatar bereits gesehen wurden
local already_seen = false
for _, seen_player in ipairs(mem.seen_lernende) do
    if seen_player == player then
        already_seen = true
        break
    end
end

if already_seen then

-- Gruppe finden, zu der ein Avatar gehört
local player_group = nil
for groupname, members in pairs(mem.groups) do
    for _, member in ipairs(members) do
        if member == player then
            player_group = groupname
            break
        end
    end
    if player_group then break end -- Schleife verlassen, falls Avatar
gefunden wurde
end

-- Gruppe auf Bildschirm anzeigen
if player_group then
    digiline_send("screen", player .. " ist in Gruppe " .. player_group)
else
    digiline_send("screen", "Fehler: " .. player .. " in keiner Gruppe
gefunden.")
end
else
    assignPlayer(player)
end
else
    digiline_send("screen", mem.hint_eineperson)
end
interrupt(mem.cooldown, "idle")
end

```

## Hilfreiche Tipps

Schütze die Geräte, damit niemand die Herausforderungen oder die Programmierung der LUA-Controller verändern kann! Dazu eignen sich der Störschutzblock und Stahltüren, die nur die Person öffnen kann, die sie auch gebaut hat.

Du kannst die TeamMatrix und die ChallengeBox zurücksetzen, indem du diese mit der rechten Maustaste anklickst und auf Execute klickst. Beachte dabei, dass bisherige Zuordnungen dabei

verloren gehen!

From:

<https://wiki.blockalot.de/> - **BLOCKALOT Wiki**

Permanent link:

<https://wiki.blockalot.de/paedagogik:challengebox-und-teammatrix>

Last update: **2025/01/23 14:57**

