



DungeonBots

Team Undead Pixels

Stewart Charles
Wesley Oates
Kevin Parker
Ken Richard



Overview

DungeonBots is the coding education game.

Fills multiple functions

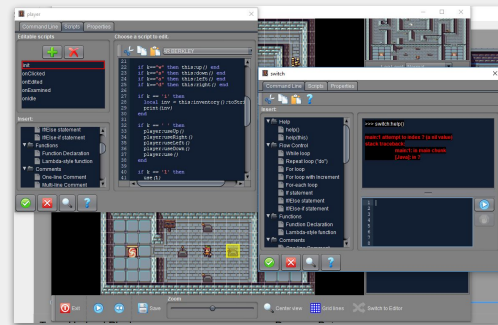
Graphically stylish and interesting

A curriculum delivery vehicle

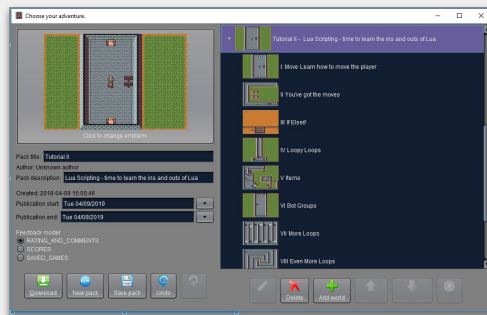
Is it a level? Is it a lesson?



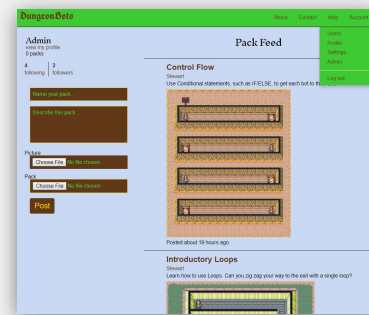
A game and game engine



An IDE



Lesson plans



A community

System Design

Game

Created our own game custom game engine with Java 8

Advanced Lua scripting integrations using Java reflection and annotations

Highly concurrent architecture

Created our own Level Editor from scratch

Website

Back end: Ruby-on-Rails

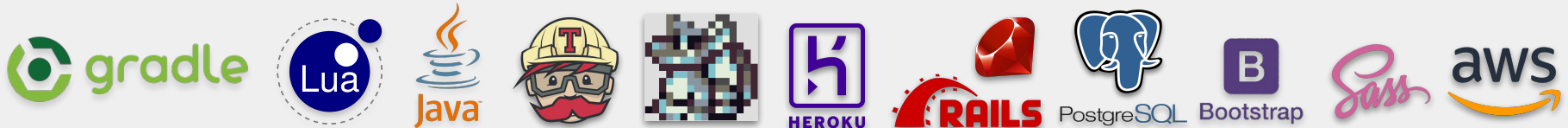
Front end: Bootstrap, SASS

Database: PostgreSQL

Development: Cloud9

Deployment: Heroku, AWS S3

API: Custom



System Capabilities - Game

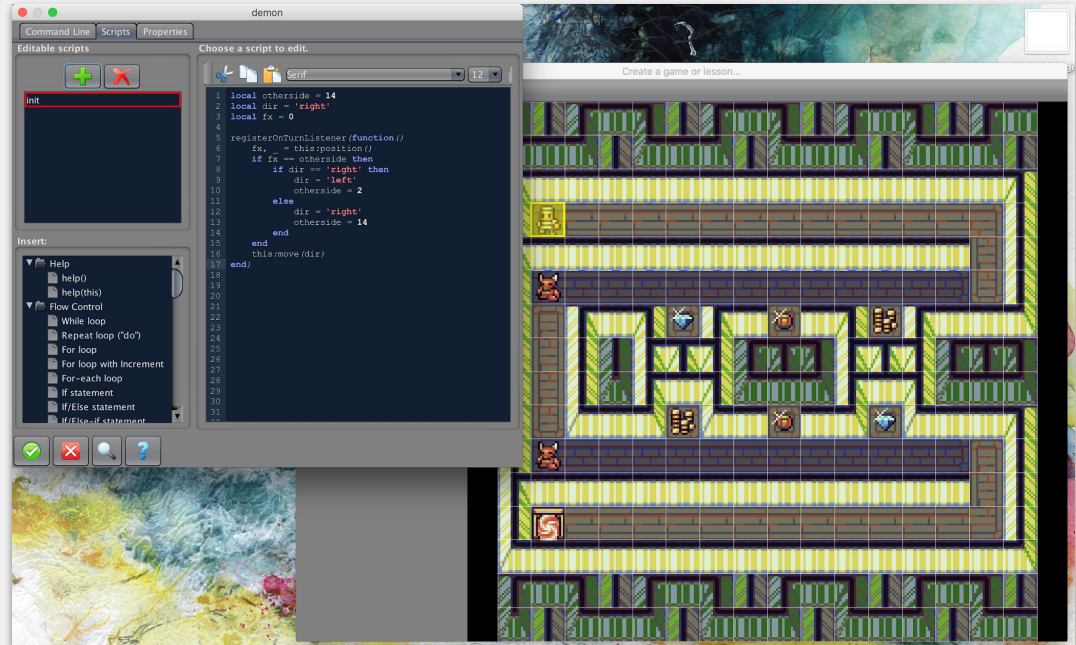
GUI level editor

Lua script editor

Lua grammar / syntax tool

Download levels from website in client

Author-defined help



System Capabilities - Website

Upload & download Level Packs

Pack Feed of user content

Help content for in-game commands

Normal user features, security

Leave us your feedback!

The screenshot displays the DungeonBots website interface. On the left, a user profile for 'Admin' is shown with 4 following and 2 followers. Below the profile are two 'Pack' management sections, each with a 'Choose File' button and a 'No file chosen' message, and a 'Post' button. The main content area on the right is divided into three sections: 'Help', 'Treasure', and 'Entity'. The 'Treasure' section contains a black box with the text 'Treasures are items that have high monetary value.'. The 'Entity' section features a 'Methods' table with columns for Name, Role, Description, and Arguments. Below the table is a 'ResponseQuestion' section with a black box containing text about its usage. At the bottom, another 'Methods' table is visible, listing methods like 'new', 'getDescription', and 'getResponseQuestions'.

DungeonBots

Admin
view my profile
0 packs

4 following | 2 followers

Name your pack ...
Describe this pack ...

Picture
Choose File No file chosen

Pack
Choose File No file chosen

Post

Help

Treasure

Treasures are items that have high monetary value.

Entity

Name	Role	Description	Arguments
getName	NONE	Get the Name of the Entity in its world	
setName	AUTHOR	Set the Name of the Entity	LuaValue
getId	AUTHOR	The Unique ID of the Entity	

ResponseQuestion

When invoked, a ResponseQuestion presents the user with a Dialog window of consisting of questions and text input responses that the user can provide and submit.

Name	Role	Description	Arguments
new	DEFAULT	Create a new Response Question Item	LuaValue The world the ResponseQuestion belongs to LuaValue A description of the Response Question LuaValue A ListTable of Questions that will each have their own responses
getDescription	AUTHOR	Gets a String representation of the Question and currently input solutions.	
getResponseQuestions	AUTHOR	Get the Response questions the User submitted.	

System Limitations

Supports deletion of entities, but no built-in combat system

Large built-in library of tiles and entities, but doesn't support custom assets

Has GUI for generating code, but no autocompletion

Pack size limitations on website

Unique Features

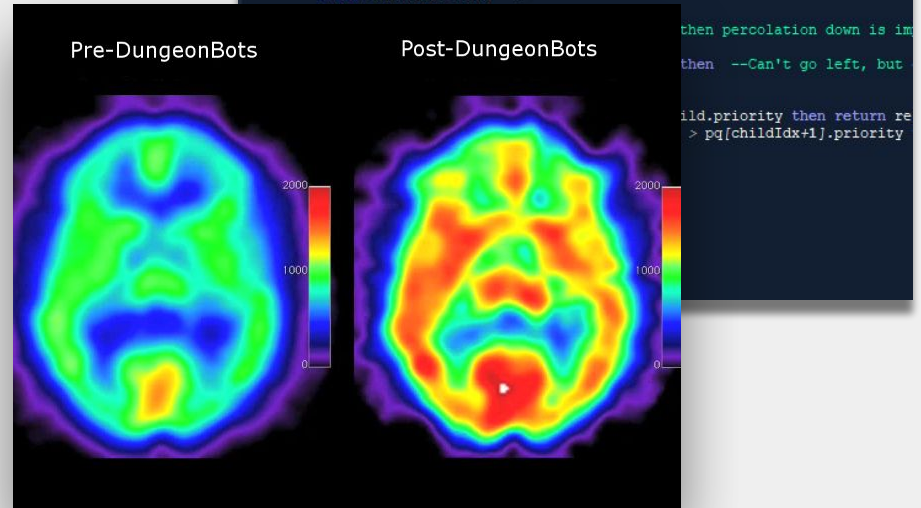
There's nothing like this out there on the market

Learn to program by playing a game

Sandboxed coding environment per entity - every entity has its own "brain"

Endlessly customizable

```
function PriorityQueue.push(item)
  local idx = #pq+1
  pq[idx] = item
  --Percolate up
  local parentIdx = intDivideBy2(idx)
  while parentIdx >= 1 do
    --player:say("idx:" .. idx .. " parentIdx:" .. parentIdx)
    if pq[parentIdx].priority <= pq[idx].priority then break end
    pq[idx] = pq[parentIdx]
    pq[parentIdx] = item
    idx = parentIdx
    parentIdx = intDivideBy2(idx)
  end
end
function PriorityQueue.pop()
  local result = pq[1]
  pq[1] = pq[#pq]
  pq[#pq] = nil
  --Percolate down
  local idx = 1
  while true do
    local childIdx = idx * 2
```



System Demo

