extends Area2D

const N = 1

const E = 2

const S = 4

const W = 8

onready var numOfEnemies = enemyloc.size()

onready var questionPopup = get\_parent().get\_node("Questions")

var cell\_walls = {Vector2(0, -1): N, Vector2(1, 0): E, Vector2(0, 1): S, Vector2(-1, 0): W}

var tile\_size = 64

var width = 16

var height = 9

var enemyloc = []

onready var EnemySprite = $Sprite

onready var Map = $TileMap

onready var Player = get\_parent().get\_node("Player")

func \_ready():

randomize()

tile\_size = Map.cell\_size

make\_maze()

generate\_enemies()

func \_physics\_process(delta):

if enemyloc.size() == 0:

set\_physics\_process(false)

pass

var pos = Player.get\_position()

var tile = Map.world\_to\_map(pos)

for k in enemyloc:

if (tile == k):

questionPopup.popup()

get\_parent().get\_node("TimerPopup").popup()

enemyloc.remove(enemyloc.bsearch(k))

var children = get\_children()

for c in children:

if c is StaticBody2D:

if Map.world\_to\_map(c.position) == tile:

remove\_child(c)

numOfEnemies = enemyloc.size()

func check\_neighbors(cell, unvisited):

var list = []

for n in cell\_walls.keys():

if cell + n in unvisited:

list.append(cell + n)

return list

func make\_maze():

var unvisited = []

var stack = []

Map.clear()

for x in range(width):

for y in range(height):

unvisited.append(Vector2(x, y))

Map.set\_cellv(Vector2(x, y), N|E|S|W)

var current = Vector2(0, 0)

unvisited.erase(current)

while unvisited:

var neighbors = check\_neighbors(current, unvisited)

if neighbors.size() > 0:

var next = neighbors[randi() % neighbors.size()]

stack.append(current)

var dir = next - current

var current\_walls = Map.get\_cellv(current) - cell\_walls[dir]

var next\_walls = Map.get\_cellv(next) - cell\_walls[-dir]

Map.set\_cellv(current, current\_walls)

Map.set\_cellv(next, next\_walls)

current = next

unvisited.erase(current)

elif stack:

current = stack.pop\_back()

pass

func shuffleList(list):

var shuffledList = []

var indexList = range(list.size())

for i in range(list.size()):

var x = randi()%indexList.size()

shuffledList.append(list[indexList[x]])

indexList.remove(x)

return shuffledList

func make\_enemy(k):

var item = preload("res://bug.png")

var node = StaticBody2D.new()

var shape = RectangleShape2D.new()

shape.set\_extents(Vector2(64,64))

var sprite = Sprite.new()

var pos = Map.map\_to\_world(k)

pos += Vector2(32,32)

node.set\_position(pos)

node.add\_child(sprite)

sprite.texture = item

node.set\_name("Enemy" + str(k))

add\_child(node)

func generate\_enemies():

var nodelist = []

for x in width:

for y in height:

nodelist.append(Vector2(x,y))

nodelist.pop\_front()

for x in range(6):

nodelist = shuffleList(nodelist)

var pos = nodelist.pop\_front()

enemyloc.append(pos)

for k in enemyloc:

make\_enemy(k)

enemyloc.sort()