extends Node2D

onready var gameTimer = $TimerPopup/Timer

onready var Player = $Player

onready var numEnemies = $Map.enemyloc.size()

func \_ready():

SoundManager.stop\_bgm("themeSong")

SoundManager.play\_me("res://assets/sounds/gameTheme.ogg",true)

root.set\_screen\_orientation(0)

get\_tree().set\_auto\_accept\_quit(false)

$TimerPopup/bg.modulate = Color(1,1,1,0.5)

loadQuestions()

$timePowerupQuantityLbl.clear()

$timePowerupQuantityLbl.append\_bbcode(userInventoryModel.getQuantityByIdx(1))

$timePowerup.connect("pressed",self,"timeUp")

gameModel.setNumQuestions(numEnemies)

pass

func loadQuestions():

$timePowerup.hide()

$timePowerupQuantityLbl.hide()

$Ready.play("ready")

#var apiUrl = gameModel.getBaseURL() + levelModel.getLevelIDByIdx(levelModel.selectedLevelIdx) + "/" + worldModel.getWorldIDbyIdx(worldModel.selectWorldIdx)

var apiUrl = gameModel.getBaseURL() + "5/World1"

apiController.apiCallGet(apiUrl)

yield(apiController, "request\_completed")

for question in apiController.getResult():

var difficulty = int(question['difficulty'])

var idx = difficulty - 1

gameModel.setAllQuestionsByDifficulty(idx,question)

func \_on\_Ready\_animation\_finished():

$Ready.queue\_free()

$TimerPopup.popup()

$TimerPopup/Timer/ms.start()

$timePowerup.show()

$timePowerupQuantityLbl.show()

Player.set\_physics\_process(true)

pass

func \_on\_Timer\_no\_time():

updateUserInventory()

$Questions.hide()

gameTimer.set\_process(false)

$TimerPopup/Timer/ms.stop()

$TimerPopup/bg.hide()

$TimeUpPopup.popup()

Player.set\_physics\_process(false)

pass

func timeUp():

if(userInventoryModel.getQuantityByIdx(1) == '0'):

pass

else:

gameTimer.s += 20

if(gameTimer.s >= 60):

gameTimer.m += 1

gameTimer.s -= 60

userInventoryModel.reducePowerupQuantyByIdx(1)

$timePowerupQuantityLbl.clear()

$timePowerupQuantityLbl.append\_bbcode(userInventoryModel.getQuantityByIdx(1))

pass

func \_on\_Questions\_popup\_hide():

if(gameModel.numQuestions == 0):

$TimerPopup/Timer/ms.stop()

Player.set\_physics\_process(false)

gameModel.calculateScore(gameTimer.m,gameTimer.s,gameTimer.ms)

updateUserInventory()

yield(apiController,"request\_completed")

$gameClearPopup.popup()

pass

func updateUserInventory():

var apiUrl = userInventoryModel.getBaseUrl() + "/gameInventory/" + userModel.getUserId()

var data = {"power1Quantity": int(userInventoryModel.getQuantityByIdx(0)),"power2Quantity": int(userInventoryModel.getQuantityByIdx(1))}

apiController.apiCallPut(data,apiUrl)