extends Popup

var enemy

var randomQuestionIdx

var currQuestion

var correctOption

var questions

var buttonsToHide = []

func \_ready():

connect("about\_to\_show",self,"loadCurrentQuestion")

for enemy in $Enemies.get\_children():

enemy.hide()

for button in $questionOptions.get\_children():

button.connect("pressed",self,"handleSelectOption",[button])

button.hide()

$ffPowerUpLbl.clear()

$ffPowerUpLbl.append\_bbcode(userInventoryModel.getQuantityByIdx(0))

$ffPowerUp.connect("pressed",self,"useFiftyfifty")

pass

func loadCurrentQuestion():

#print("Current Difficulty" + str(gameModel.currDifficulty))

$currentDiffLbl.clear()

$currentDiffLbl.append\_bbcode("Current Difficulty: " + str(gameModel.currDifficulty))

buttonsToHide = []

for item in self.get\_children():

item.show()

for item in $questionOptionsLbl.get\_children():

item.show()

$correctWrongAni.hide()

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

loadEnemy()

$avatarAni.play(userModel.getUserAvatar())

questions = gameModel.getQuestionsByDifficulty()

randomize()

randomQuestionIdx = int(rand\_range(0,questions.size()))

currQuestion = questions[randomQuestionIdx]

correctOption = int(currQuestion["correctOption"])

print("Correct Answer: " + str(correctOption))

setQuestionTitleTxt()

setQuestionOptions()

setQuestionSelectButtons()

pass

func loadEnemy():

randomize()

var randomEnemyIndex = ceil(rand\_range(-1, 2))

for item in $Enemies.get\_children():

if(item.get\_index() == randomEnemyIndex):

enemy = item

enemy.show()

enemy.play("idle")

else:

item.hide()

func setQuestionTitleTxt():

$questionTitleLbl.clear()

$questionTitleLbl.append\_bbcode("[center]" + currQuestion['questionTitle'])

func setQuestionOptions():

for item in $questionOptionsLbl.get\_children():

var itemIdx = item.get\_index()

var optionNumber = "option" + str(itemIdx+1)

item.clear()

item.append\_bbcode("[center]" + currQuestion[optionNumber])

pass

func handleSelectOption(button):

highlightCorrectOption()

disableQuestionSelectButtons()

gameModel.removeQuestionFromDifficulty(randomQuestionIdx)

var selectedIdx = button.get\_index()

if(selectedIdx == correctOption):

handleCorrect()

else:

handleWrong()

func setQuestionSelectButtons():

for button in $questionOptions.get\_children():

button.show()

button.set\_block\_signals(false)

func disableQuestionSelectButtons():

for button in $questionOptions.get\_children():

button.set\_block\_signals(true)

func highlightCorrectOption():

for button in $questionOptions.get\_children():

if(button.get\_index() == correctOption):

button.modulate = Color(0,1,0)

else:

button.modulate = Color(1,0,0)

func handleCorrect():

enemy.frame = 1

enemy.play("down")

gameModel.increaseDifficulty()

playCorrectWrongAni("correct")

pass

func handleWrong():

enemy.frame = 1

enemy.play("swing")

gameModel.decreaseDifficulty()

playCorrectWrongAni("wrong")

pass

func playCorrectWrongAni(value):

$correctWrongAni.frame = 1

$correctWrongAni.show()

$correctWrongAni.play(value)

func resetAll():

for button in $questionOptions.get\_children():

button.modulate = Color(1, 1,1,1)

for item in self.get\_children():

item.hide()

gameModel.decreaseQuestions()

self.hide()

pass

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

resetAll()

pass

func useFiftyfifty():

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

pass

else:

var numButtons = $questionOptions.get\_child\_count()

for i in range(numButtons):

if(i != correctOption):

buttonsToHide.insert(buttonsToHide.size(),i)

var itemToRemove = int(rand\_range(0,buttonsToHide.size()))

buttonsToHide.remove(itemToRemove)

if(buttonsToHide.size() == 2):

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

$questionOptions.get\_child(buttonsToHide[i]).hide()

$questionOptionsLbl.get\_child(buttonsToHide[i]).hide()

userInventoryModel.reducePowerupQuantyByIdx(0)

$ffPowerUpLbl.clear()

$ffPowerUpLbl.append\_bbcode(userInventoryModel.getQuantityByIdx(0))

pass