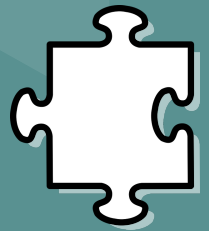
Final Presentation: Puzzle



By:

Eduardo Velazco
Hasitha Gampa
Tania Hernandez
Melissa Iraheta

Introduction

Purpose

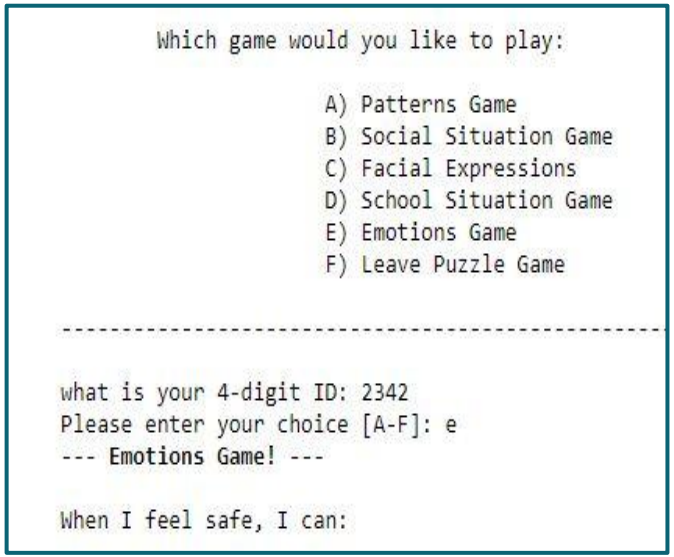
- Target Group: Children aged 9 to 13 years old.
- Goal: Assess school-aged minors (9-13 years old) for signs of autistic symptoms.
- Implementation: Can then be used accordingly to adjust therapy treatments.

Game Rules/Measures

- 5 categories
  - 4-8 questions each with 3-4 answer choices each
- Reward points for questions answered correctly
  - Track points for each category.
  - Puzzle image
- We will be measuring the progress of each category.
  - This will help identify any particular categories that the child is having trouble with.

Puzzle Game Chatbot

Chatbot Menu



- User:
  - Enters a 4- digit user ID
  - Chooses game category
  - Answers questions for that category

Game Code

Emotions Game Code

```
elif choice == "E" or choice == "e":
    print( "\033[1m"+ "--- Emotions Game! ---"+"\033[0m")
    class e_quest:
        def __init__(self, Emotion_qs, images, acceptable_answers, correct_answer, points):
            self.Emotion_qs = Emotion_qs
            self.images = images
            self.acceptable_answers = acceptable_answers
            self.correct_answer = correct_answer #correct answer
            self.points = points
                #add correct answer
    Emotion_qs = [
                e_quest("When I feel grumpy, I can: ", Image.open("Images/Emotion qs/grumpyEmoji.jpg"),["Throw
                e_quest("When I feel worried, I can: ", Image.open("Images/Emotion qs/anxiousEmoji.jpg"), ["Co
                e_quest("When I feel bored, I can: ", Image.open("Images/Emotion qs/boredEmoji.jpg"), ["Play a
                e_quest("When I feel caring, I can: ", Image.open("Images/Emotion qs/lovingEmoji.jpg"), ["Shou
                e_quest("When I feel curious, I can: ", Image.open("Images/Emotion qs/curiousEmoji.jpg"), ["As
                e_quest("When I feel safe, I can: ", Image.open("Images/Emotion qs/safeEmoji.jpg"), ["Relax",
                e_quest("When I feel jealous, I can: ", Image.open("Images/Emotion qs/jealousEmoji.jpg"), ["Li
                e_quest("When I feel lazy, I can: ", Image.open("Images/Emotion qs/lazyEmoji.jpg"), ["Procrast
            ]

    Emotion_a = []
    e_test = random.shuffle(Emotion_qs)

    user_answers = {"Incorrect Answers":0,"Invalid Answers":0, "Correct Answers":0}
    def run_emotions(e_test):
        global activities_passed
        global user_answers
        score = 0
```

```
    #add the dictionary here to store user answer#
    for question_all in Emotion_qs: #asks user each question
        #user_answer = input(question_all.questions).lower().strip()
        print("")
        print(question_all.Emotion_qs)
        q = [x for x in range(len(question_all.acceptable_answers))]
        random.shuffle(q)
        #iterate through all answers and display image
        for image in question_all.acceptable_answers:
            get_image = np.asarray(question_all.images)
            image_plot = plt.imshow(get_image)
            plt.xticks([])
            plt.yticks([])
        plt.show()
        q = [x for x in range(len(question_all.acceptable_answers))]
        random.shuffle(q)
        correct_answer = q.index(question_all.correct_answer)
        for ix in range(len(question_all.acceptable_answers)):
            print("({:s}) {:s}".format("ABCD"[ix],question_all.acceptable_answers[q[i
        not_answer = True
        while not_answer:
            sel = input("Your answer:").lower().strip()
            if len(sel) != 1:
                print("answers should be 1 character")
                user_answers["Invalid Answers"] += 1
```

```
with open('puzzle_Escores.csv', 'a', newline='') as out_file:
    writer = csv.writer(out_file)
    #writer.writerow(["Date", "User ID", "Correct Answers", "Incorrect Answers", "Invalid Answers"])
    writer.writerow([timestamp,userID, user_answers["Correct Answers"], user_answers["Incorrect Answers"], u
```

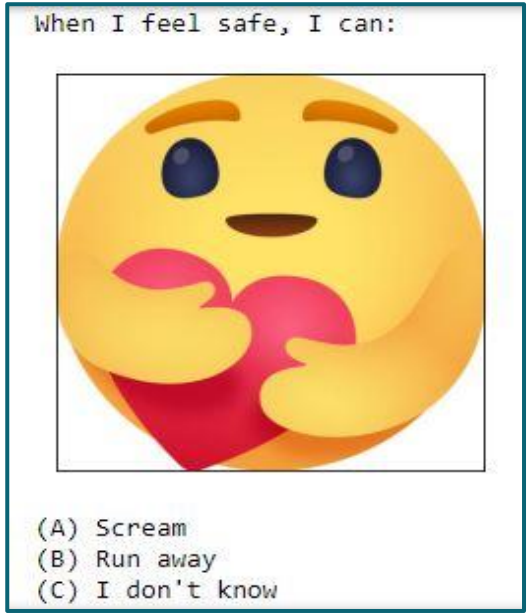
Check Answers

Emotions Game Code

```
    plt.yticks([])
plt.show()

q = [x for x in range(len(question_all.ok_answers))] #runs through answers
random.shuffle(q)
correct_answer = q.index(question_all.correct_answer)
for ix in range(len(question_all.ok_answers)):
    print("({:s}) {:s}".format("ABCD"[ix],question_all.ok_answers[q[ix]]))
not_answer = True
while not_answer:
    sel = input("Your answer is: ").lower().strip()
    if len(sel) != 1:
        print("Please, answers should be 1 character (ex:a,b,c or d)")
        user_answers["Invalid Answers"] += 1
        continue
    if not (sel in "abcd"):
        print("Oops, that's not an option!")
        user_answers["Invalid Answers"] += 1
        continue
    if sel == "abcd"[correct_answer]:
        print("Great job! You get one point!")
        not_answer = False
        score += 1
        user_answers["Correct Answers"] += 1
        scores["Patterns"] += 1
    else:
        print("Nope, better luck next time!")
        user_answers["Incorrect Answers"] += 1
        not_answer = False
```

Emotions Game Output



Reward System: Puzzle Image

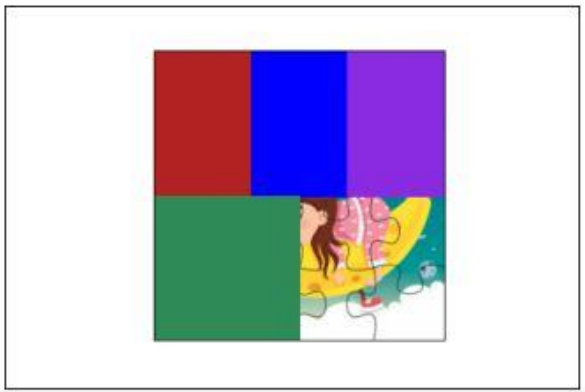
Puzzle Image Code

```
    # checks to see if user meets criteria
        if user_answers["Correct Answers"] >= 6:
            activities_passed += 1
            #shows image
            plt.imshow(prize_image[activities_passed])
            plt.xticks([])
            plt.yticks([])
            plt.show()
        elif user_answers["Correct Answers"] <= 5:
            print("Sorry! Could not unlock a puzzle piece. Try again!")

    run_emotions(e_test)
    user_answers
    scores
    loop = True
    print()
```

Puzzle Image Output

```
Your answer:a
Great job! You get one point!
Your score is: 7/8
```



User Testing & Analytics

User Answers

```
#read in files for each game category
E_scores <- read_csv("puzzle_Escores.csv")
FE_scores <- read_csv("puzzle_FExscores.csv")
Social_scores <- read_csv("Social_Qscores.csv")
School_scores <- read_csv("puzzle_SSscores.csv")
P_scores <- read_csv("puzzle_patternsscores.csv")
```

```
#convert each file to a dataframe
E_scores <- data.frame(E_scores)
E_scores
FE_scores <- data.frame(FE_scores)
FE_scores
School_scores <- data.frame(School_scores)
School_scores
Social_scores <- data.frame(Social_scores)
Social_scores
P_scores <- data.frame(P_scores)
P_scores
```

Emotions Game User Answers

Date	User.ID	Correct.Answers	Incorrect.Answers	Invalid.Answers
<chr>	<chr>	<dbl>	<dbl>	<dbl>
05/17/2021	4567	2	6	1
05/17/2021	7770	3	5	0
05/17/2021	7771	4	4	0
05/17/2021	7772	3	5	0
05/17/2021	7771	2	6	0
05/17/2021	7771	1	7	0
05/17/2021	7773	7	1	0
05/17/2021	7774	5	3	0
05/17/2021	7775	6	2	0

Overall Scores

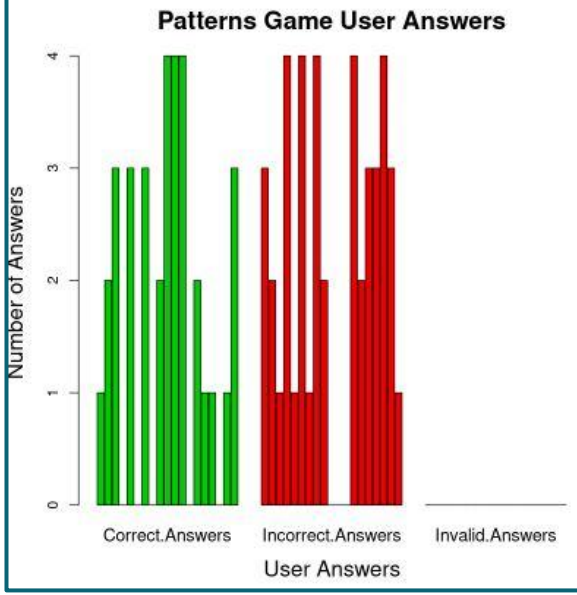
```
correct_emotions <- round(mean(E_scores$Correct.Answers),2)
correct_fe <- round(mean(FE_scores$Correct.Answers, 2))
correct_school <- round(mean(School_scores$Correct.Answers, 2))
correct_social <- round(mean(Social_scores$Correct.Answers, 2))
correct_pattern <- round(mean(P_scores$Correct.Answers, 2))
game_categories <- c("Emotions", "Facial Expressions", "School Situations", "Social Skills", "Patterns")
mean_correct <- c(correct_emotions,correct_fe, correct_school, correct_social, correct_pattern)
scores_per_category <- data.frame("Game Category" = game_categories, "Mean Correct Answers" = mean_correct, "Total Possit
scores <- round((scores_per_category$Mean.Correct.Answers / scores_per_category$Total.Possible.Correct.Answers),2)
scores_per_category$Mean.Scores <- scores
scores_per_category
```

Game.Category	Mean.Correct.Answers	Total.Possible.Correct.Answers	Mean.Scores
<chr>	<dbl>	<dbl>	<dbl>
Emotions	3.74	8	0.47
Facial Expressions	4.00	8	0.50
School Situations	3.00	5	0.60
Social Skills	4.00	9	0.44
Patterns	2.00	4	0.50

Analytics- Patterns Game

```r
colors = c("green3","red2", 'yellow3') #green for correct, red for incorrect, and yellow for invalid answers

barplot(as.matrix(P_scores[,c('Correct.Answers','Incorrect.Answers','Invalid.Answers')]), beside = TRUE, col= rep(color
```




```r
#Users that unlocked a puzzle piece for patterns game
P_scores[P_scores$Correct.Answers >= 3,]
```

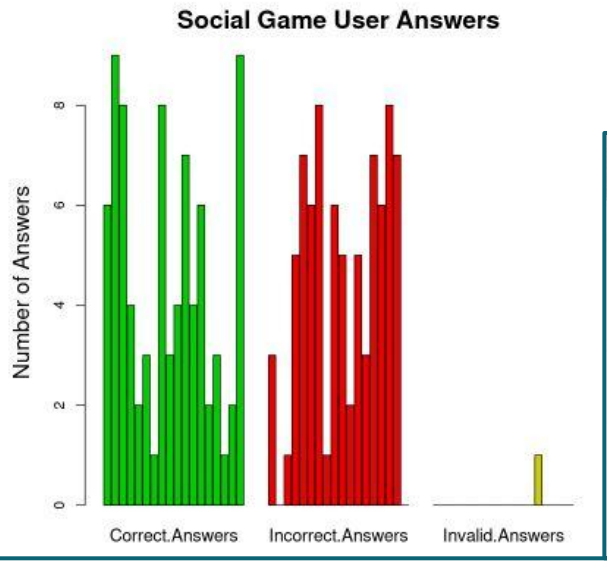
A data.frame: 7 × 5

	Date	User.ID	Correct.Answers	Incorrect.Answers	Invalid.Answers
	<chr>	<chr>	<dbl>	<dbl>	<dbl>
3	05/17/2021	7771	3	1	0
5	05/17/2021	7773	3	1	0
7	05/17/2021	7775	3	1	0
10	05/17/2021	1234	4	0	0
11	05/17/2021	2345	4	0	0
12	05/17/2021	4567	4	0	0
19	05/17/2021	0512	3	1	0

Analytics- Social Situations

```r
#bar graphs for the Social Game
colors = c("green3","red2", 'yellow3') #green for correct, red for incorrect, and yellow for invalid answers

barplot(as.matrix(Social_scores[,c('Correct.Answers','Incorrect.Answers','Invalid.Answers')]), beside = TRUE, col=
```




```r
#Users that unlocked a puzzle piece for social situations game
Social_scores[Social_scores$Correct.Answers >= 7,]
```

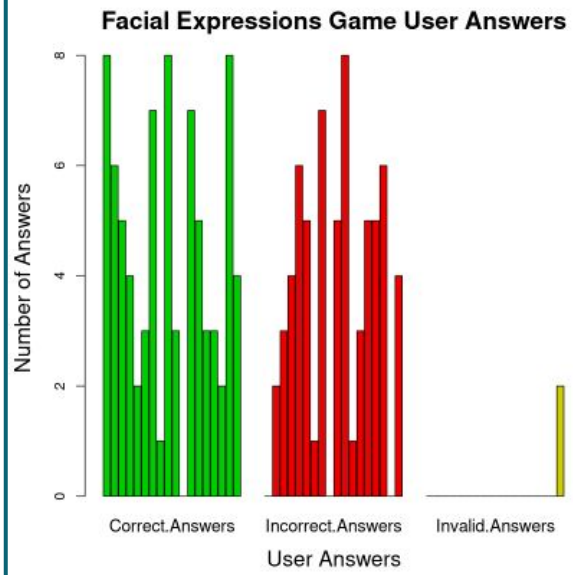
A data.frame: 5 × 5

	Date	User.ID	Correct.Answers	Incorrect.Answers	Invalid.Answers
	<chr>	<chr>	<dbl>	<dbl>	<dbl>
2	05/17/2021	7770	9	0	0
3	05/17/2021	7771	8	1	0
8	05/17/2021	7775	8	1	0
11	05/17/2021	2345	7	2	0
18	05/17/2021	0512	9	0	0

Analytics- Facial Expressions

```
#bar graphs
colors = c("green3","red2", 'yellow3') #green for correct, red for incorrect, and yellow for invalid answers

barplot(as.matrix(FE_scores[,c('Correct.Answers','Incorrect.Answers','Invalid.Answers')]), beside = TRUE, col= rep(colors, ea
```



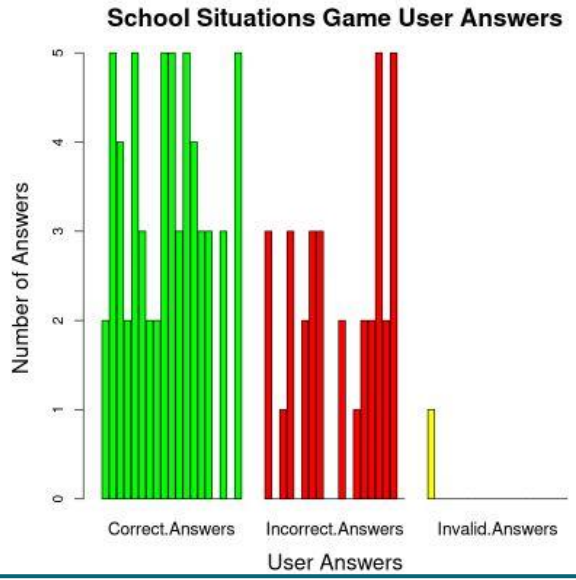

```
#Users that unlocked a puzzle piece for facial expressions game
FE_scores[FE_scores$Correct.Answers >= 6,]
```

A data.frame: 6 × 5

	Date	User.ID	Correct.Answers	Incorrect.Answers	Invalid.Answers
	<chr>	<chr>	<dbl>	<dbl>	<dbl>
1	05/17/2021	4567	8	0	0
2	05/17/2021	7770	6	2	0
7	05/17/2021	7775	7	1	0
9	05/17/2021	1234	8	0	0
12	05/17/2021	4567	7	1	0
17	05/17/2021	0512	8	0	0

Analytics- School Situations

```
1  colors = c("green","red", 'yellow') #green for correct, red for incorrect, and yellow for invalid answers
2
3  barplot(as.matrix(School_scores[,c('Correct.Answers','Incorrect.Answers','Invalid.Answers')]), beside = TRUE, col=
```




```
#Users that unlocked a puzzle piece for school situations game
School_scores[School_scores$Correct.Answers >= 4,]
```

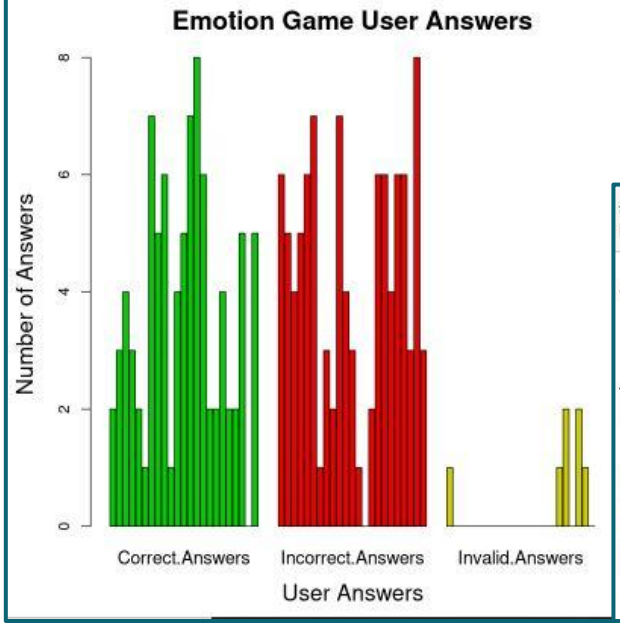
A data.frame: 8 × 5

	Date	User.ID	Correct.Answers	Incorrect.Answers	Invalid.Answers
	<chr>	<chr>	<dbl>	<dbl>	<dbl>
2	05/17/2021	7770	5	0	0
3	05/17/2021	7771	4	1	0
5	05/17/2021	7773	5	0	0
9	05/17/2021	1234	5	0	0
10	05/17/2021	1234	5	0	0
12	05/17/2021	4567	5	0	0
13	05/17/2021	4567	4	1	0
19	05/17/2021	0512	5	0	0

Analytics- Emotion Game

```
colors = c("green3","red2", "yellow3") #green for correct, red for incorrect, and yellow for invalid answers

barplot(as.matrix(E_scores[,c('Correct.Answers','Incorrect.Answers','Invalid.Answers')]), beside = TRUE, col
```




```
#Users that unlocked a puzzle piece for emotion game
E_scores[E_scores$Correct.Answers >= 6,]
```

A data.frame: 6 × 5

	Date	User.ID	Correct.Answers	Incorrect.Answers	Invalid.Answers
	<chr>	<chr>	<dbl>	<dbl>	<dbl>
7	05/17/2021	7773	7	1	0
9	05/17/2021	7775	6	2	0
13	05/17/2021	2345	7	1	0
14	05/17/2021	4567	8	0	0
15	05/17/2021	5678	6	2	0
25	05/19/2021	2342	7	1	0

Thank you

Any Questions ?