

Rubric for Grading the Project

CRITERIA	EXEMPLARY (90-100)	SATISFACTORY (80-89)	DEVELOPING (70-79)	BEGINNING (below 70)	WEIGHT
Experimental Plan (Flowchart/ Algorithm) (SO-PI: BI)	Experimental plan has supporting details and diagram/algorith hm that is stated and well explained	Experimental plan has supporting details and diagram/algorith m that is stated but not explained	Experimental plan is vague or brief. It has supporting details and does not have diagram/ algorithm	No experimental plan presented	20%
Codes/Data/ Program	Data is well utilized in the program. Program code are easy to read. Program output has no error. Questions are answered completely and correctly	Data is somewhat utilized in the program. Program code are easy to read. Program output has an output but logically incorrect. Some questions are answered completely and correctly	Data is not utilized in the program. It has a missing Significant code/syntax in the program.	No program presented	30%
	Appropriate	Appropriate	Appropriate	Appropriate	10%

Use of Appropriate Tools and Techniques <i>(SO-PI: K1)</i>	tools and techniques are properly used for all aspects of the project	tools and techniques are used in most of the aspects of the project and all of these are used properly	tools and techniques are used in majority of the aspects of the project and all of these are used properly	tools and techniques are used in less than half of the aspects of the project and/or tools are not used properly in at least half the aspects of the project	
Project Documentation	Project documentation is orderly presented starting from statement of the problem, to objective of the project, followed by review of literature, design consideration, presentation of data or output and Conclusion. The report was grammatically correct, logically presented and used the required format.	Project documentation is complete with statement of the problem, objectives, design consideration, presentation of data and output and conclusion. The report had minimal grammatical errors and somewhat presented logically. The required format was used.	Project documentation is basically limited to algorithm presentation of data and output but no basis of the design was presented. The report had a lot of grammatical errors and not logically presented; the required format was barely used.	Project documentation is not reflective of algorithm design and/or characterization. The report had a lot of grammatical errors, was not logically presented and the required format was not used.	30 %

Project Presentation SO-PI: G2	Project presentation is complete and backed up by complete Design consideration, logic formulation and review of related literature	Project presentation is complete with algorithm simulation results backed up by design considerations.	Project presentation shows a system completely simulated but is not backed up by clear explanation of how algorithm was derived	Project presentation lacks clarity in terms of presenting and characterizing the behavior of the algorithm	10 %
TOTAL					100%

DE LA SALLE UNIVERSITY - MANILA

DICER: A mix of poker and dices

A Term Project

Presented to Mr. RUIZ, RAMON STEPHEN L. , Mr. VELASCO, NEIL OLIVER M.

In Partial Fulfillment of the

Requirements for the Course PROLOGI & LBYCPA1

By

TAN, CIELBERT L.
BALTAZAR, ANDREI SEBASTIAN PANLAQUE
EQ2, EQ4

April, 2023

Table of Contents

I.	Rubric for Grading the Project	1-3
II.	Title Page	4
III.	Table of Contents	5
IV.	Introduction	6
V.	Review of Related Literature	9
VI.	Methodology	9
VII.	Results	
VIII.	Discussion of Results	
IX.	Analysis, Conclusion	
X.	References	
XI.	Appendix	

I. Introduction

A dice game where five dice will be used. There is a scoring system for the points and what makes this unique is the difference in scoring the points. The goal of the game is to get 10,000. If you get a score below the minus it will go to the negative number the limit of the negative number is -1000. There would also be a bit of the poker rule where it has full-house, straight-flush, two-pair, etc.

A. Background of the Study

Dice games have been around for centuries, with evidence of their use dating back to ancient civilizations. These games are often simple to learn, yet challenging to master, making them a popular choice for both casual and serious gamers. In recent years, the popularity of dice games has grown significantly, with new variations and rules being developed to keep the game interesting and engaging (American Psychologist, 2014).

One of the latest variations of the traditional dice game is a scoring system that incorporates both positive and negative point values, as well as unique scoring opportunities for certain dice combinations. This scoring system adds a new level of strategy and excitement to the game, as players must balance their desire for high scores with the risk of rolling negative point values. (Trends in Cognitive Sciences, 2012)

Despite the popularity of this new variation, there has been little research done on the game's impact on players' cognitive and decision-making skills. Additionally, there is a lack of information on the optimal strategies for maximizing point values and minimizing risk.

Therefore, the purpose of this study is to investigate the effects of playing the unique scoring system dice game on cognitive and decision-making skills. The study will explore the game's impact on players' risk-taking behavior, strategic thinking, and pattern recognition abilities. Furthermore, the study aims to identify the most effective strategies for maximizing point values while minimizing the risk of negative scores.

The results of this study will provide valuable insights into the cognitive and decision-making processes involved in playing this unique scoring system dice game. Additionally, the study will contribute to the development of strategies and best practices for playing the game, which can benefit both casual and serious gamers alike.

B. Problem Statement

The ideal scenario for any game is that it stimulates players' cognitive and decision-making skills, providing a fun and engaging experience. Unfortunately, this is not always the case with dice games, where little research has been done on the cognitive processes involved. In the unique scoring system dice game we are investigating, players risk negative scores if they do not choose their moves wisely. The reality is that without a thorough understanding of the game's intricacies, players may struggle to achieve high scores, leading to frustration and disinterest in the game. The consequences of this lack of knowledge can be significant, as players may miss out on the opportunity to improve their cognitive skills while enjoying a fun and challenging game. Therefore, it is imperative to conduct research on the cognitive and decision-making skills required to play the unique scoring system dice game, and identify the most effective strategies for maximizing point values while minimizing risk. By doing so, we can ensure that players have a positive experience, improve their cognitive skills, and enhance their enjoyment of the game. (Information Systems Journal, 2016)

C. Objectives

C.1 General Objective

1. Develop a scoring system based on the combination of dice rolls
2. Implement the rules of the game, including the negative limit and the goal of 10,000 points
3. Design a user-friendly interface to display the score and game status
4. Ensure the program can handle user input and display the appropriate output
5. Test the program to identify and fix any bugs or errors

C.2 Specific Objectives

1. Create a pseudocode or program code that implements the specific rules and mechanics of the dice game, including the addition and subtraction of points based on dice rolls, as well as the rules for full house, straight flush, two pairs, and three of a kind combinations.
2. Develop a user-friendly graphical interface that displays the current game status, including the player's score and the dice combinations rolled in each round.
3. Ensure that the program handles user input effectively, including rolling the dice and selecting the appropriate combinations to add or subtract points.
4. Test the program thoroughly to identify and fix any bugs or errors, ensuring that it runs smoothly and provides a seamless gaming experience for players.
5. Provide clear instructions or a user manual to guide players through the rules and mechanics of the game, making it accessible to both novice and experienced players.

D. Significance of the Project

The dice game project offers several benefits to players. First, it provides an enjoyable form of entertainment that can be played alone or with others. The game's unique scoring system, combined with the poker-like rules, offers a challenging and engaging experience for players of all skill levels. Furthermore, it offers an excellent opportunity to spend quality time with friends and family while fostering social interaction and improving communication skills.

Secondly, the game can be used as a learning tool to improve mathematical skills, such as probability and addition and subtraction of integers. Players can develop their mental arithmetic skills by calculating the points earned or lost after each roll of the dice. This can be especially beneficial for children who are learning math, as it provides a fun and interactive way to practice and develop their skills.

Finally, the dice game project can be an effective stress-reliever for individuals who need a break from their daily routine. The game's simple yet challenging mechanics can help players to relax and unwind while also providing an opportunity to exercise their cognitive skills. Overall, this project has several benefits that make it a worthwhile endeavor for both players and developers alike.

II. Review of Related Literature

This review explores the various cognitive biases that affect players' decision-making in poker. It discusses how factors such as risk aversion, the framing effect, and the sunk cost fallacy can impact a player's game, and how awareness of these biases can lead to better decision-making and improved outcomes. (Toplak, Jain, & Flora , 2015)

This review traces the history of dice games from ancient civilizations to modern-day variations. It explores the cultural significance of dice games, their role in gambling, and the evolution of the games themselves. The review also delves into the various types of dice games and their rules, highlighting their unique features and appeal. (Scarne, 1962)

This review examines the role of probability and strategy in dice games. It discusses the importance of understanding probability and statistics when playing dice games, and how this knowledge can inform strategy and increase the likelihood of winning. The review also explores different types of strategies, including aggressive and conservative approaches, and how they can be applied in various dice games. (Hogarth & Karelaia, 2005)

III. Methodology

A. Conceptual Framework – IPO Chart

Input:

The input for this project would be the user's input of rolling the five dice. The user can also choose which dice to keep and which ones to roll again. The user's score will also be considered as input for the program.

Process:

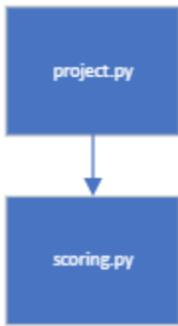
The program will evaluate the user's input and perform the following tasks:

1. Calculate the score based on the combination of dice rolls (full-house, straight-flush, two-pair, etc.)
2. Check if the score is above or below the negative number limit
3. Check if the user has reached the goal of 10,000 points
4. Display the score and game status to the user
5. Repeat the process until the user reaches the goal or goes below the negative limit

Output:

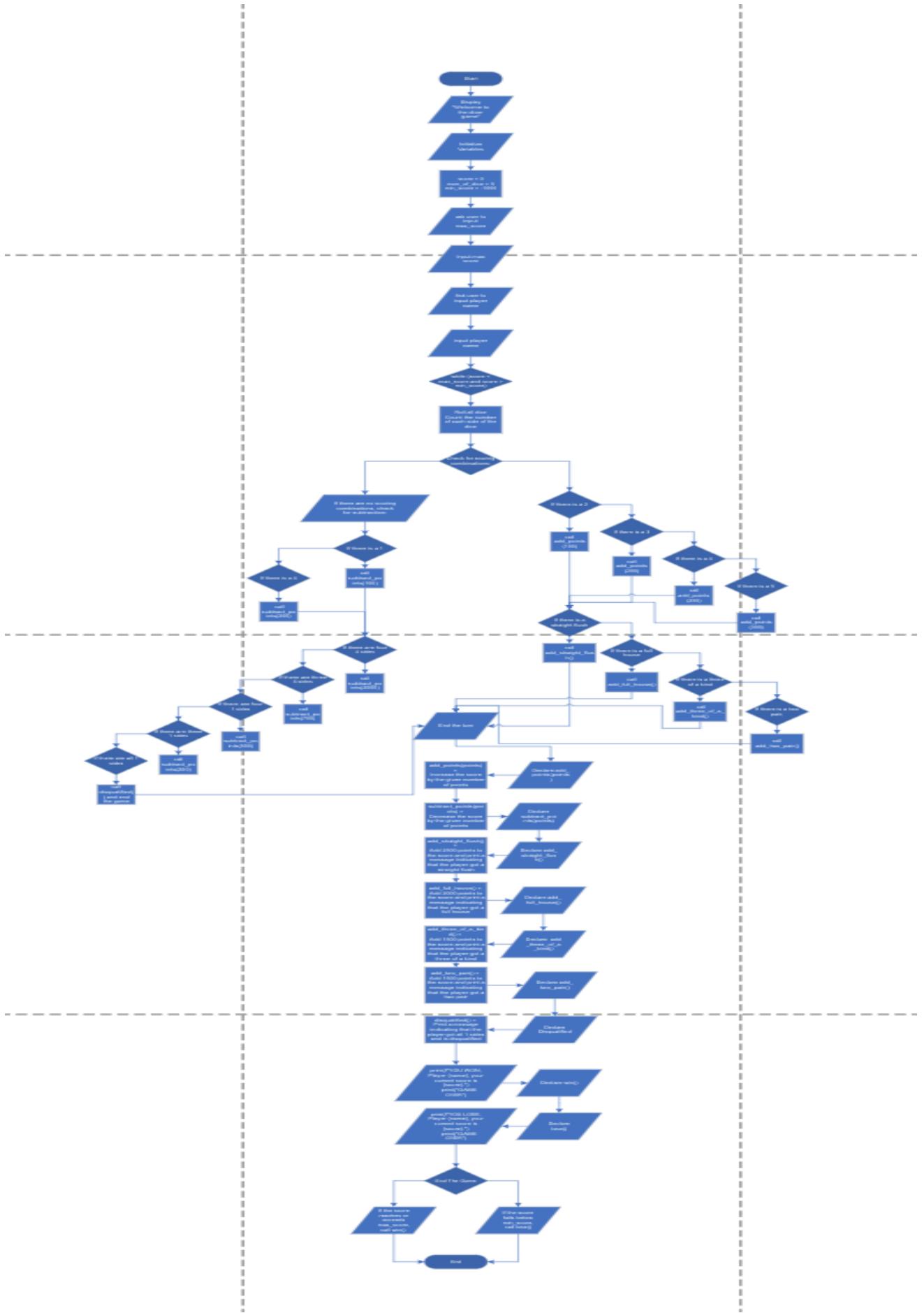
The output of the program would be the score and game status, which includes the current points, whether the user has reached the goal, or has gone below the negative limit. The program will also display the options for the user to roll the dice and choose which dice to keep.

B. Hierarchy Chart



C. Flowchart

Link To the flowchart: [CPA1 Prolog Project Documentation Flowchart](#)



D. Pseudocode

1. START
2. Display “Welcone to the dicer game”
3. Declare Player name
4. name = input("What is your name? ")
5. Declare max_score
6. Max_score = "What is the desired Limit of your score?")
7. Initialize variables
 - a. score = 0
 - b. num_of_dice = 5
 - c. max_score
 - i. Ask user to input max score
 - ii. Max_score = input of user for their max score
 - d. min_score = -1000
8. while (score < max_score and score > min_score)
 - a. Roll all dice
 - b. Count the number of each side of the dice
 - c. Check for scoring combinations:
 - d. If there is a 2, call add_points(100)
 - e. If there is a 3, call add_points(200)
 - f. If there is a 5, call add_points(250)
 - g. If there is a 6, call add_points(300)
 - h. If there is a straight flush, call add_straight_flush() and end the turn
 - i. If there is a full house, call add_full_house() and end the turn
 - j. If there is a three of a kind, call add_three_of_a_kind() and end the turn
 - k. If there is a two pair, call add_two_pair() and end the turn
 - l. If there are no scoring combinations, check for subtraction:
 - i. If there is a 4, call subtract_points(200)
 - ii. If there is a 1, call subtract_points(100)

- iii. If there are four 4 sides, call subtract_points(2000)
 - iv. If there are three 4 sides, call subtract_points(750)
 - v. If there are four 1 sides, call subtract_points(500)
 - vi. If there are three 1 sides, call subtract_points(200)
 - vii. If there are all 1 sides, call disqualified() and end the game
 - m. End the turn
9. add_points(points)
- a. Increase the score by the given number of points
10. subtract_points(points)
- a. Decrease the score by the given number of points
11. add_straight_flush()
- a. Add 2500 points to the score and print a message indicating that the player got a straight flush
12. add_full_house()
- a. Add 2000 points to the score and print a message indicating that the player got a full house
13. add_three_of_a_kind()
- a. Add 1500 points to the score and print a message indicating that the player got a three of a kind
14. add_two_pair()
- a. Add 1500 points to the score and print a message indicating that the player got a two pair
15. disqualified()
- a. Print a message indicating that the player got all 1 sides and is disqualified
16. win()
- a. Print a message indicating that the player reached or exceeded the maximum score and won the game
17. lose()
- a. Print a message indicating that the player fell below the minimum score and lost the game
18. End of game

- a. If the score reaches or exceeds max_score, call win() and end the game
 - i. Display “You Won the Game”
- b. If the score falls below min_score, call lose() and end the game
 - i. Display “You Lost the Game”

19. END

IV. Results

```
What is your name? Andrei Baltazar
What is the desired Limit of your score?3000
Round 1
Your dice rolls: [1, 4, 1, 1, 6]
Which dice do you want to keep (enter space-separated indices, starting from 1)? 5
Your new dice rolls: [6, 6, 6, 3, 3]
Andrei Baltazar, you got a three-of-a-kind with 6s and a pair! You scored 1500 points.
Andrei Baltazar, you got a pair! You scored 750 points.
Andrei Baltazar, your current score is 2250.
Andrei Baltazar, you got 300 points for rolling a 6.
Andrei Baltazar, your current score is 2550.
Andrei Baltazar, you got 300 points for rolling a 6.
Andrei Baltazar, your current score is 2850.
Andrei Baltazar, you got 300 points for rolling a 6.
Andrei Baltazar, your current score is 3150.
Andrei Baltazar, you got 200 points for rolling a 3.
Andrei Baltazar, your current score is 3350.
Andrei Baltazar, you got 200 points for rolling a 3.
Andrei Baltazar, your current score is 3550.
YOU WON, Player Andrei Baltazar, your current score is 3550.
GAME OVER
```

```

Do you want to play again? (y/n) y
Round 1
Your dice rolls: [5, 5, 2, 5, 3]
Which dice do you want to keep (enter space-separated indices, starting from 1)? 3
Your new dice rolls: [2, 3, 1, 6, 6]
Andrei Baltazar, you got a pair! You scored 750 points.
Andrei Baltazar, your current score is 750.
Andrei Baltazar, you got 100 points for rolling a 2.
Andrei Baltazar, your current score is 850.
Andrei Baltazar, you got 200 points for rolling a 3.
Andrei Baltazar, your current score is 1050.
Andrei Baltazar, you lost 1000 points for rolling a 1.
Andrei Baltazar, your current score is 50.
Andrei Baltazar, you got 300 points for rolling a 6.
Andrei Baltazar, your current score is 350.
Andrei Baltazar, you got 300 points for rolling a 6.
Andrei Baltazar, your current score is 650.
Round 2
Your dice rolls: [3, 4, 2, 1, 1]
Which dice do you want to keep (enter space-separated indices, starting from 1)? 2 4 5
Your new dice rolls: [4, 1, 1, 3, 1]
Andrei Baltazar, your current score is 0.
Andrei Baltazar, you lost 750 points for rolling a 4.
Andrei Baltazar, your current score is -750.
Andrei Baltazar, you lost 1000 points for rolling a 1.
Andrei Baltazar, your current score is -1750.
Andrei Baltazar, you lost 1000 points for rolling a 1.
Andrei Baltazar, your current score is -2750.
Andrei Baltazar, you got 200 points for rolling a 3.
Andrei Baltazar, your current score is -2550.
Andrei Baltazar, you lost 1000 points for rolling a 1.
Andrei Baltazar, your current score is -3550.
YOU LOSE, Player Andrei Baltazar, your current score is -3550.
GAME OVER
Do you want to play again? (y/n) █

```

```

Round 1
Your dice rolls: [6, 1, 5, 6, 1]
Which dice do you want to keep (enter space-separated indices, starting from 1)? 1 3 4
Your new dice rolls: [6, 5, 6, 2, 1]
Andrei Baltazar, you got a pair! You scored 750 points.
Andrei Baltazar, your current score is 750.
Andrei Baltazar, you got 300 points for rolling a 6.
Andrei Baltazar, your current score is 1050.
Andrei Baltazar, you got 250 points for rolling a 5.
Andrei Baltazar, your current score is 1300.
Andrei Baltazar, you got 200 points for rolling a 6.
Andrei Baltazar, your current score is 1600.
Andrei Baltazar, you got 100 points for rolling a 2.
Andrei Baltazar, your current score is 1700.
Andrei Baltazar, you lost 1000 points for rolling a 1.
Andrei Baltazar, your current score is 700.
Round 2
Your dice rolls: [3, 4, 5, 5, 1]
Which dice do you want to keep (enter space-separated indices, starting from 1)? 1 3 4
Your new dice rolls: [3, 5, 6, 1]
Andrei Baltazar, you got a pair! You scored 750 points.
Andrei Baltazar, your current score is 750.
Andrei Baltazar, you got 200 points for rolling a 3.
Andrei Baltazar, your current score is 950.
Andrei Baltazar, you got 250 points for rolling a 5.
Andrei Baltazar, your current score is 1200.
Andrei Baltazar, you got 250 points for rolling a 5.
Andrei Baltazar, your current score is 1450.
Andrei Baltazar, you lost 1000 points for rolling a 1.
Andrei Baltazar, your current score is 450.
Andrei Baltazar, you got 300 points for rolling a 6.
Andrei Baltazar, your current score is 750.
Round 3
Your dice rolls: [2, 5, 1, 3, 3]
Which dice do you want to keep (enter space-separated indices, starting from 1)? 1 2 4 5
Your new dice rolls: [2, 5, 3, 3, 1]
Andrei Baltazar, you got a pair! You scored 750 points.
Andrei Baltazar, your current score is 750.
Andrei Baltazar, you got 100 points for rolling a 2.
Andrei Baltazar, your current score is 850.
Andrei Baltazar, you got 250 points for rolling a 5.
Andrei Baltazar, your current score is 1100.
Andrei Baltazar, you got 200 points for rolling a 3.
Andrei Baltazar, your current score is 1300.
Andrei Baltazar, you got 200 points for rolling a 3.
Andrei Baltazar, your current score is 1500.
Andrei Baltazar, you lost 1000 points for rolling a 1.
Andrei Baltazar, your current score is 500.
Round 4
Your dice rolls: [5, 5, 5, 5, 6]
Which dice do you want to keep (enter space-separated indices, starting from 1)? 1 2 3 4 5
Your new dice rolls: [5, 5, 5, 5, 6]
Andrei Baltazar, you got a four-of-a-kind with 5s! You scored 2000 points.
Andrei Baltazar, your current score is 2000.
Andrei Baltazar, you got 250 points for rolling a 5.
Andrei Baltazar, your current score is 2250.
Andrei Baltazar, you got 250 points for rolling a 5.
Andrei Baltazar, your current score is 2500.
Andrei Baltazar, you got 250 points for rolling a 5.
Andrei Baltazar, your current score is 2750.
Andrei Baltazar, you got 250 points for rolling a 5.
Andrei Baltazar, your current score is 3000.
Andrei Baltazar, you got 300 points for rolling a 6.
Andrei Baltazar, your current score is 3300.
YOU WON, Player Andrei Baltazar, your current score is 3300.
GAME OVER
Do you want to play again? (y/n) █

```

V. Discussion of Results

It will show first the introduction messages then it will move on to the question where it will ask for your name and goal limit. Afterwards, the game will start, in the first game the rolled dice are 1,4,1,1,6, since 6 is the only good number we will only keep 6. We will put position 5 as

the input for the box and it will roll 4 dice again, now the new dice are 6,6,6,3,3. It will then compute for the score which is a total of 3550, since we set the limit to 3000, we won.

Now the game will ask whether we want to continue to play again or quit the program. In the second game, the goal will still be the same, 3000. The game will roll again and the process will be the same, but in the second game we lost since we got too many deductions. In the third game, we won again.

VI. Analysis, Conclusion and Future Directives

Analysis:

The dice game presented above is an interesting combination of luck and strategy. It involves rolling five dice and using a scoring system that rewards certain combinations of numbers while penalizing others. Additionally, there are special combinations such as a straight flush, full house, three of a kind, and two pair, which provide players with extra points. The game is designed to be won by reaching a score of 10,000 or more, with the limit of negative score being -1,000.

One interesting feature of this game is the use of colored dice, where 4 and 1 have a different color. This feature adds an additional layer of complexity and strategy, as players need to consider the possibility of rolling a 1 or 4 and the corresponding negative points.

Another unique feature is the poker-like combinations that can be rolled, which provide players with additional points. These combinations add a level of excitement to the game, as players try to roll the right combination to gain an advantage.

Conclusion:

Overall, the dice game presented above is a well-designed and entertaining game that combines luck and strategy. The scoring system is balanced and provides players with various opportunities to gain points while also avoiding negative points. The inclusion of special combinations adds excitement and strategy to the game, making it more engaging for players.

Future Directives:

One potential improvement for this game could be to add more special combinations, such as a four of a kind or a five of a kind. This would provide players with even more opportunities to gain extra points and make the game more varied.

Another possible improvement could be to introduce more complex scoring rules, such as bonus points for rolling certain combinations in a specific order or for rolling a certain number of combinations in a single turn. These rules would add an additional layer of strategy and increase the difficulty of the game.

Finally, the game could be adapted into a mobile app or online game, allowing players to play against each other from anywhere in the world. This would expand the game's reach and make it more accessible to a wider audience.

References

- Granic, I., Lobel, A., & Engels, R. C. M. E. (2014). The benefits of playing video games. *American Psychologist*, 69(1), 66–78. <https://doi.org/10.1037/a0034857>

Bavelier, D., Green, C. S., & Schute, M. G. (2012). Cognitive benefits of playing video games. *Trends in Cognitive Sciences*, 16(7), 408–416.

<https://doi.org/10.1016/j.tics.2012.06.003>

Shade, M. K., Fox, C., & Hansen, A. N. (2019). The cognitive benefits of board games for adults: A review and discussion of modern research. *Games*, 10(4), 37.

<https://doi.org/10.3390/g10040037>

Weinmann, M., Basten, D., & Johnsen, D. (2016). Improving decision-making through gamification: The influence of leaderboards on the effectiveness of decision support systems. *Information Systems Journal*, 26(2), 149–175.

<https://doi.org/10.1111/isj.12087>

Forneris, C. A., O'Rourke, R. J., & Clapp, A. E. (2018). Game-based cognitive behavioral therapy: A model for treating children with anxiety in group settings. *Journal of Psychosocial Nursing and Mental Health Services*, 56(6), 41–45.

<https://doi.org/10.3928/02793695-20180327-02>

Toplak, M. E., Jain, R., & Flora, D. B. (2015). The influence of cognitive biases on decision making in a simulated poker game. *Applied Cognitive Psychology*, 29(6), 883-893. doi: 10.1002/acp.317

Scarne, J. (1962). Scarne on Dice. New York: Crown Publishers.

Hogarth, R. M., & Karelaia, N. (2005). "Ignorance and Surprise: Science, Probability, and Strategy in Simple Games." *Cognitive Science*, 29(1), 1-19. doi: 10.1207/s15516709cog2901_1

Appendices

A. User's Manual

First, start the program, then when the program asks , type your name and what limit goal would you like. The game will start once you enter both these information. When the first roll of dice appears, the program will ask which dice do you want to keep. You must enter the position of the dice you want to keep with space in between, an example will be: 2 4 5. This means I would like to keep the 2nd, 4th, and 5th dice. The program will roll the dice that you don't want again. After rolling it will display the new dice and automatically compute your points, then you can keep on playing till you hit below -1000 or reach the targeted goal.

B. Source Code

Project.py module

```
import random
```

```
import scoring
```

```
# welcome message
```

```
print("Welcome to DICER, the thrilling game that combines the excitement of poker and the chance of rolling the dice! In this game, you will use five dice to score points and reach your limit.")
```

```
print("What sets DICER apart is its unique scoring system, which offers a variety of ways to earn points. And just like in poker, you'll have the chance to make classic combinations like full-house, straight-flush, and two-pair.")
```

```
print("But beware, if you score below the limit, you'll end up in the negative, with a minimum of -1000 points. So, you'll need to use your skill and luck to avoid losing big.")
```

```
print("Get ready to roll the dice and test your luck in DICER!")
```

```
# get player's name
```

```
name = input("What is your name? ")
```

```
limit = int(input("What is the desired Limit of your score?"))
```

```
while True:
```

```
    # initialize score and loop until user reaches goal or goes below negative limit
```

```
    score = 0
```

```
    rounds = 1
```

```

while -1000 <= score < limit:
    # roll dice
    dice_rolls = [random.randint(1, 6) for _ in range(5)]

    # display dice rolls
    print(f"Round {rounds}")
    print("Your dice rolls: ", dice_rolls)

    # ask which dice to keep
    keep_dice_str = input("Which dice do you want to keep (enter space-separated
indices, starting from 1)? ")
    keep_dice = [int(idx) - 1 for idx in keep_dice_str.split()]
    kept_dice = [dice_rolls[idx] for idx in keep_dice]

    # roll remaining dice
    remaining_dice = [dice_rolls[idx] for idx in range(5) if idx not in keep_dice]
    new_dice_rolls = [random.randint(1, 6) for _ in range(len(remaining_dice))]
    dice_rolls = kept_dice + new_dice_rolls

    # display new dice rolls
    print("Your new dice rolls: ", dice_rolls)
    score = scoring.dice_score(dice_rolls, name, limit)
    rounds += 1

play_again = input("Do you want to play again? (y/n) ")
if play_again.lower() == "n":
    Break

```

Scoring.py module

```

def dice_score(dice_rolls, name, limit):
    #global name, score

```

```

score = 0

dice_counts = {}
for value in dice_rolls:
    dice_counts[value] = dice_counts.get(value, 0) + 1

# check for straight flush
if set(dice_rolls) == {1, 2, 3, 4, 5} or set(dice_rolls) == {2, 3, 4, 5, 6}:
    score += 2500
    print(f'{name}, you got a straight flush! You scored 2500 points.')

# check for 5 of a kind
elif any(count == 5 for count in dice_counts.values()):
    value = next(key for key, value in dice_counts.items() if value == 5)
    score += 2 * value
    print(f'{name}, you got five-of-a-kind with {value}s! You scored {2 * value} points.')

# check for 5 of a kind of 4
elif 4 in dice_counts and dice_counts[4] == 4:
    score -= 2000
    print(f'{name}, you lost 2000 points for rolling four-of-a-kind with 4s.')

# check for 5 of a kind of 1
elif 1 in dice_counts and dice_counts[1] == 5:
    score -= 5000
    print(f'{name}, you lost 5000 points for rolling five-of-a-kind with 1s.')

else:
    # calculate score based on individual dice values
    for value, count in dice_counts.items():
        if count == 4 and value not in [1, 4]:
            score += 2000
            print(f'{name}, you got a four-of-a-kind with {value}s! You scored 2000 points.')
        elif count == 3 and value not in [1, 4] and any(count == 2 for count in
dice_counts.values()):

```

```

score += 1500
print(f" {name}, you got a three-of-a-kind with {value}s and a pair! You scored 1500
points.")

elif count == 2:
    pairs = [v for v, c in dice_counts.items() if c == 2 and v not in [1, 4]]
    if len(pairs) >= 1:
        score += 750
        print(f" {name}, you got a pair! You scored 750 points.")
    elif value in [1, 4]:
        score -= 500
        print(f" {name}, you lost 500 points for rolling a pair of {value}s.")
    print(f" {name}, your current score is {score}.")

for dice in dice_rolls:
    if dice == 1:
        score -= 1000
        print(f" {name}, you lost 1000 points for rolling a 1.")
    elif dice == 4:
        score -= 750
        print(f" {name}, you lost 750 points for rolling a 4.")
    elif dice == 2:
        score += 100
        print(f" {name}, you got 100 points for rolling a 2.")
    elif dice == 3:
        score += 200
        print(f" {name}, you got 200 points for rolling a 3.")
    elif dice == 5:
        score += 250
        print(f" {name}, you got 250 points for rolling a 5.")
    elif dice == 6:
        score += 300

```

```

print(f'{name}, you got 300 points for rolling a 6.')

print(f'{name}, your current score is {score}.')
# display current score
if score <= -1000:
    print(f'YOU LOSE, Player {name}, your current score is {score}.')
    print("GAME OVER")
elif score >= limit:
    print(f'YOU WON, Player {name}, your current score is {score}.')
    print("GAME OVER")
return score

```

C. Work breakdown		
Student Name	Tasks Assigned	Percentage of the Work Contribution
Tan, Cielbert 	<ul style="list-style-type: none"> ● Code Module 2(scoring.py) ● Video Editor ● Document ● 	50%

C. Work breakdown

Baltazar, Andrei 	<ul style="list-style-type: none">• Code Module 1(Project.py)• Video Demo• Document• 	50%
---	--	-----