Comprehensive Exercise Presentation
Group: The Bunco Bunch

By: Lliam Rankins, William Krajcovic, Shiv Patel and Micah Tucker

Project Summary

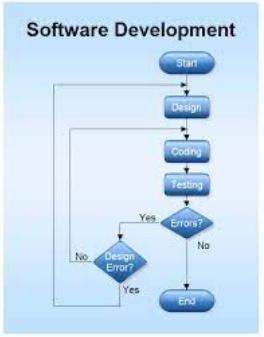
- Our Task: Coding Bunco in Java
- Prompting User in Terminal
- 2-10 Players
- Round Based
- Dice Rolling
- Scoring

Process

- Defining Requirements
- Software Design
- Implementation
- Testing
- Documentation

List of Requirements

- Have two players
- Be able to roll three dice
- All a seed to be used
- Check if dice match target value
- Check for big bunko
- Check for little bunko
- Add scores properly
- Check if player reaches 21
- Add win message
- Allows the player to roll dice
- Allows player to start a new game



Comprehensive Exercise Directory

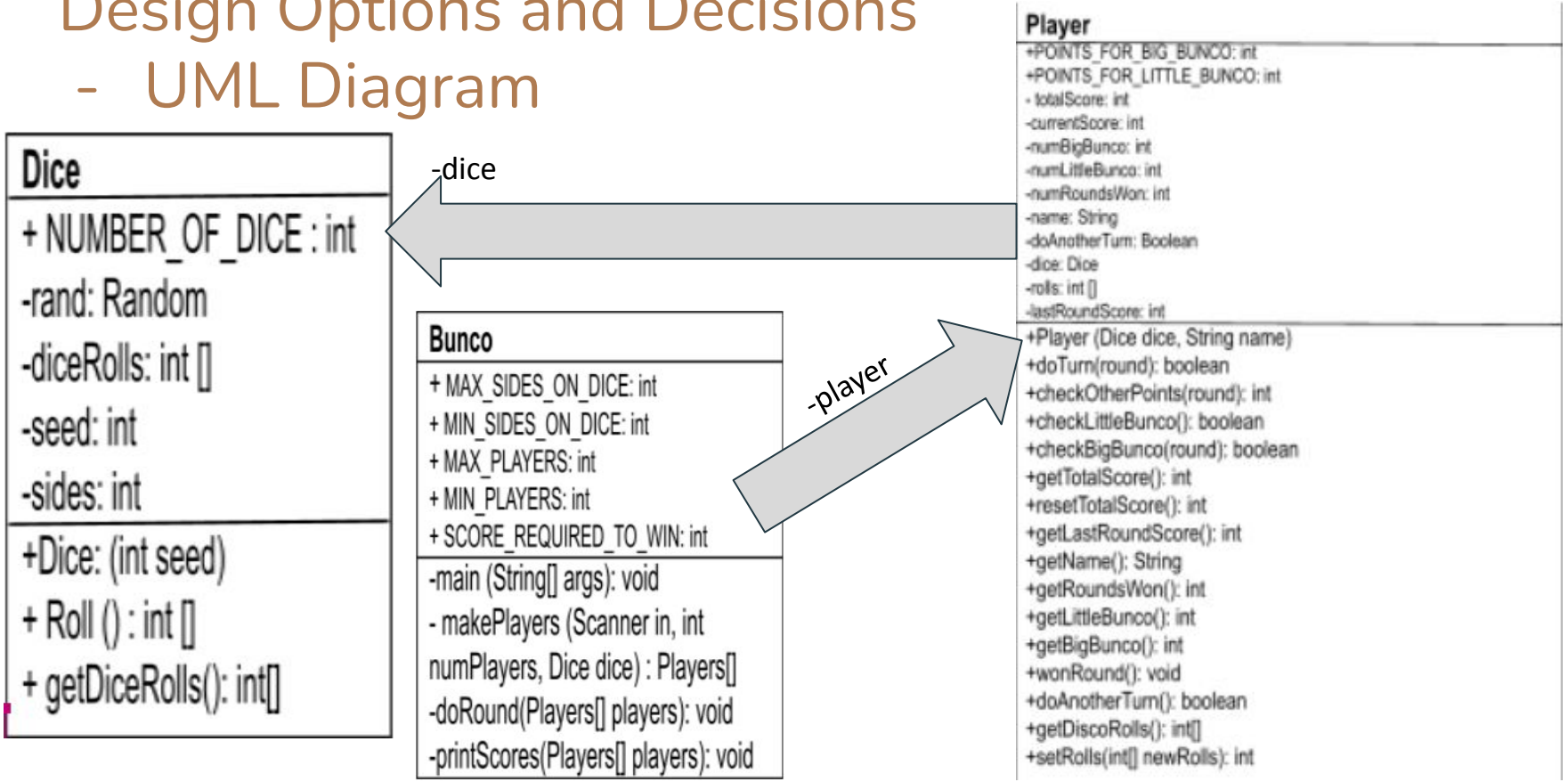
-> CE
  -> src
    -> Bunco.java
    -> Player.java
    -> Dice.java
  -> test
    -> DiceTest.java
    -> PlayerTest.java
  -> lib
    -> junit-platform-console-standalone-1.6.2.jar
  -> bin
    -> Bunco.class
    -> Dice.class
    -> Player.class
  -> project_docs
    -> SystemTestPlan_Bunco.pdf

testStartProgram	$ java -cp bin Bunco (Ctrl+C to stop execution)	There will be displayed 3 dice and a roll button, the dice will be 1, 2, 3. There will be a new game button, (Figure this out later)

Requirement Assumptions/Additions

- Functioning game of bunco, including rolling, point counting, etc.
- Initially Supported only 2 players, but later added the ability for players from to 2-10
- Uses three 6-sided dice,but later changed to support 6-26 sided dice
- Ability to see/not see points/rolls for current round (Depends on the user and what they enter)

Design Options and Decisions
- UML Diagram



Testing

- **System Testing/Bunco Program**
  - No boundary values as there are two functions
  - Equivalence Testing included testing the input of integers asking for dice sides and numbers of players.
  - System tests plan included testing the functions of the Bunco game such as testBigBuno(),testLittleBunco(), testWin(), etc.
- **Unit/Integration Testing: DiceTest**
  - Unit Testing that tested the methods in the Dice class such as getRolls() and Roll() methods
- **Unit/Integration Testing: PlayerTest**
  - Unit Testing that tested the methods in the Player class such as getName(), DoTurn(),testBigBunco(),and testLittleBunco(), testCheckOtherPoints(),These methods tested the methods and the game functions needed for Bunco

Example: System Testing

testAllSetup Authros: Billy Krajcovic, Lliam Rankins, Shiv Patel, Micah Tucker	$ java -cp bin Bunco **Enter in 6 then click enter** **Enter in 2 then click enter** **Enter in A then click enter** **Enter in B then click enter** **Enter in Y then click enter** **Enter in Y then click enter** **Enter in N then click enter**	Please enter an integer (Range 6-26): How many players are playing? Please enter an integer (Range 2-10): Player 1 name: Player 2 name Do you wish to see the dice rolls(y/n): Do you wish to see each players scores at the end of the round(y/n): A It is your turn to roll, would you like to roll? (y/n)	Please enter an integer (Range 6-26): How many players are playing? Please enter an integer (Range 2-10): Player 1 name: Player 2 name Do you wish to see the dice rolls(y/n): Do you wish to see each players scores at the end of the round(y/n): A It is your turn to roll, would you like to roll? (y/n)

Demo!

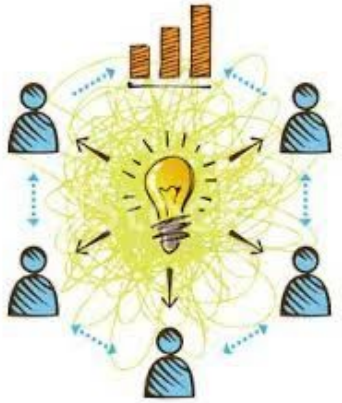
```
$ java -cp bin Bunco 1
What sided die do you want to play with?
Please enter an integer (Range 6-26): 6

How many players are playing?
Please enter an integer (Range 2-10): 2

Player 1 name: A
Player 2 name: B

Do you wish to see the dice rolls(y/n): y

Do you wish to see each players scores at the end of the round(y/n): y

A It is your turn to roll, would you like to roll? (y/n): y
A rolled: [4, 5, 2]
A:
Total Score: 0
Current Round Score: 0
Rounds Won: 0
Total Little Buncos: 0
Total Big Buncos: 0

B:
Total Score: 0
Current Round Score: 0
Rounds Won: 0
Total Little Buncos: 0
Total Big Buncos: 0

B It is your turn to roll, would you like to roll? (y/n):
```

Answering Technical Questions

- Is there any error handling within your class programs?
  - Yes, we do have error handling in our class programs. Bunco is the main one that we have error handling as it is the only program that takes in the user's input. The user will have to enter an integer in specific questions if the input is not in the range or if they enter anything that's not an integer the user will be prompted repeatedly till they enter a valid input.

What We Learned

1. Understanding all of the instructions before starting any parts of the software process steps is incredibly important. This will prevent any confusions and will help you save time. This was something we regret not having done and we definitely recommend this to others.
2. Dividing our codes in smaller chunks helped us be organized and helped us understand the concept much better.




thank you
Ins
NumLk
Del
ScrLk
Shift
PgUp
PgDn
End
Home