

10: Finishing the Game

CSCI 6626 / 4526 April 2018

1 Goals

- To implement the logic for an entire game.
- To integrate all parts of the controller: enrolling players, taking a turn, finishing the game.
- To test the interaction of the Game, Player, PlayerList, Board, and PolyDice classes.
- To add exception handling to the project.

2 Exceptions.

Define an exception class called **BadPlayer**. From it, derive two exception classes, **BadName** and **BadColor**. The derived classes will be used when the human user enters a name or a color (but not both) that is already in use. The base class will be used if both are in use. The base class should contain `char*` members to store the actual input lines the user entered.

3 Game Components

Finish your `play()` function. Modify it to play an entire game. Get the first player from the list. Repeat the following actions as long as the game status does not change from the initial state.

- If there are no players left, end the game.
- Otherwise, do one turn.
- If the current player wins on this turn, announce the result and end the game.
- Otherwise, go to the next player.

Add exception handlers. In your `getPlayers()` function, catch these exceptions: `bad_alloc`, `BadPlayer`, `BadName`, `BadColor` and the general exception (`...`). Handle these exceptions in such a way that the user is told what he entered, what is wrong with it, and he is given another chance to enter good data.

Write a private function `checkData()` in the Game class to validate the input data (both the name and the color must be unique). Throw an exception if the data is invalid. Call it from `getPlayers()` for each player input.

4 Testing

Make up an input file for the fake dice that is long enough to complete a game with 3 players. Let one player resign after the first turn; continue with the other two. In the dice-data file, provide data that will exercise stop, bust, wonColumn, and moves where one pair of dice fails but the second can be used. Let each player win one column and let one player win the game. When you run the test, you may either use the normal user interface to enter go/stop/quit commands, according to your test plan, or you may integrate these commands into your fake dice data file. Submit a test plan (a succinct list of moves and expected outcomes), and output.