Project 1: Building Pacman Game

Emanuel Calderon

John Hopkins University

Table of Contents

[Building Pacman Game 3](#_Toc66032222)

[Design and Analysis: 3](#_Toc66032223)

[TDSR 3](#_Toc66032224)

[General Programming Design 3](#_Toc66032225)

[Alternatives 5](#_Toc66032226)

[Learning Outcomes 5](#_Toc66032227)

[Wrap Up: 6](#_Toc66032228)

[References 7](#_Toc66032229)

[Appendix A 8](#_Toc66032230)

# Building Pacman Game

This writing contains the information for the solution of the Pacman game built using Java as directed by Project 1 assignment. The objective of this assignment was to utilize the skills and tools gained from the beginning of the course into creating a game and have a bit of fun doing it.

## Design and Analysis:

The program created to meet the minimum requirements of this project had its challenging moments. n addition to an extensive literature review, 40 interviews were conducted for this study. The goal of conducting interviews was to find out how students looked at the use of smartphones in the classroom.

### TDSR

Prior to jumping into task, the programmer walked through the Top-Down Stepwise Refinement (TDSR) pseudocode as to avoid any major setbacks to the coding performed. The TDSR set the foundation of the overall algorithm, made it very apparent of how repeatable the algorithm will be and assisted to identify where there is room to replace repeats into methods. Inevitably there were some backtracks needed to be made in order to get the game to be played correctly, though using the TDSR assisted to avoid the major ones.

### General Programming Design

Program organization follow as (1) Bring in scanners needed (2) Call all variables used (3) Allow user first inputs from the request display of the x and y variables (4) Create internal grid with given parameters set by user and conditions of cookies, dots and Pacman location (5) Print commands, game instructions and internal grid of Pacman, dots and 14% cookies randomly placed on grid (6) Actual functionality of the game from 0-4 command inputs by user (7) end game statistics. All other codes are methods to the flow of the project class to assist with the benefits of the use of methods.

Structure of entire code is a mixture of arrays, for’s, if’s, and switches. Within callouts, small arithmetic was used to complete desired action placed by player/user. Methods were used wherever possible as to reduce the redundancies of the algorithm, shorten huge lines of code in main body and make the code look neater. Previous code looked messy and very long to sift through. As mentioned before the TDSR assisted to identify where to replace with method callout. There were some algorithms that made it impossible to replace with methods given the number of variables needed to be returned.

Commands for 1, 2, and 3 are processed through switch cases in a loop and set to repeat wherever the user decides to reuse those commands. The command 3 was a bit tricky to perform when attempting to make the current Pacman location blank and input Pacman into the new location. Command 3 is the only command with nested if statement due to the cookie instances. Withing the movement the out of bounds logic was placed in as well. After trial and error, it seemed a simple arithmetic was needed to get the desired movements. Command 3 was to be placed in a method though methods only allow one return. Cookie if statement was the only code placed as method to reduce the redundant call outs and fully utilizing the methods design intent. Command 4 and 0 were simple enough and were also placed in methods to reduce the redundant call outs.

As mentioned before the Pacman’s state is maintained (using command 3 for example) via arithmetic and array replacements, see Figure 1. Essentially, program is being told to (1) Sift through grid (using for loop) for Pacman location and direction (2) Once found, replace current grid with blank (3) Arithmetic to the direction the Pacman is facing (4) If the location it is heading is a cookie perform the necessary task (5) place the same direction of Pacman at new location (6) count as a move (7) break out. For the commands 1 or 2 entered by the user to rotate the Pacman, similar concepts are performed to maintain Pacman state. From the aforementioned steps only (1) and (2) are used. All commands have the counter placed for statistical outputs.

### Alternatives

Other alternatives approached were very limited to the fact of the grid being placed in an array. Alternatives include the optimization of the code to clean up using the methods which was taught in the course. Although in some cases this did not work as previously mentioned. When there were more values that needed to be returned, it made methods null to use. At times a value would not return when requested which gave programmer headache until it was realized when error proofing that a counter was not being returned but the print statement was. The coding in this case was reverted back to the original formatting.

### Learning Outcomes

There were much learning outcomes from this project. (1) It further emphasized the benefits from the use of methods (2) It strengthened the confidence in understanding the use of the arrays and many benefits with the use of the array/string combination (3) Improved understanding on how to use probability with loop functions (4) Learned best grid randomization feature for future projects.

Something that could have been done differently, maybe use more of the string commands replace features to shorten up the multiple lines present with verbiage including to replace features of the Pacman. Make it look more professional than a simple stepwise function. Possibly include the math within one of the string arguments we learned in the course.

The code was completed early on which gave the programmer much time to clean up. Code that was cleaned up included the none statement present for the Pacman in the left corner. Without the if, code would always run. An example of the clean up of methods is seen in Figure 2 which helps show clean look.

## Wrap Up:

This project challenged students in this course in many ways. The skills gained whole working through this project will be very useful with future assignments and projects performed in the remainder of the course.

# References

Deal, J. (2021, 1 1). *Mini Project1 Instructions Spr2021*. Retrieved from 605.201 Mini-Project 1: https://blackboard.jhu.edu/bbcswebdav/pid-9226692-dt-content-rid-99036029\_2/courses/EN.605.201.81.SP21/Mini%20Project1InstructionsSpr2021.pdf

Purdue University. (2019, 10 1). *Purdue Online Writing Lab*. Retrieved from APA Style Introduction: https://owl.purdue.edu/owl/research\_and\_citation/apa\_style/apa\_style\_introduction.html

Schildt, H. (2019). *Java The Complet Reference Eleventh Edition.* Chicago: McGraw-Hill Education.

Streefkerk, R. (2017). *APA format for academic papers (6th edition)*. Retrieved from Scribr: https://www.scribbr.com/apa-style/6th-edition/archived-format/

# Appendix A

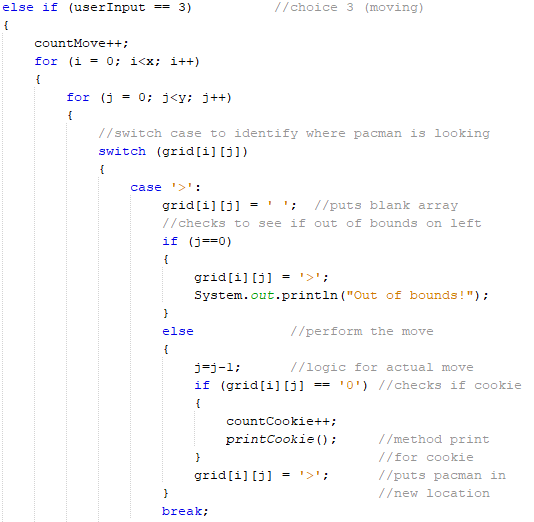


Figure 1. User input choice 3 Pacman movement

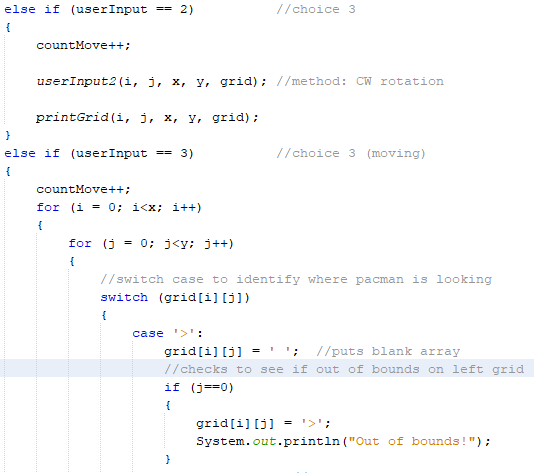


Figure 2. Differences between method and no method.