DREW WOLFE | ONID: WOLFEDR

CS 362-400 WINTER 2018
ASSIGNMENT: 2

REFACTOR

Card selection was somewhat arbitrary except that I knew I didn't want to mess around with anything other than ACTION cards. I figured that I would be able to establish a design pattern for the extracted functions, and I figured that tweaking individual actions would be easy to control as well as predicting the impact of the changes/bugs. I also went down the list of cards on dominionstrategy.com's list of cards and literally looked for the least amount of text as it seemed to translate to the lowest complexity in card actions. I figured we'll have to extract all the cards out eventually, so nailing down the design pattern seemed like a better objective than muscling through complex and potentially far-reaching actions.

As far as the specific cards, I chose the GREAT HALL, the VILLAGE (the least impressive, or interesting "village" out there), the COUNCIL ROOM, and obviously the ADVENTURER and SMITHY cards. Fortunately most, if not all variables, structs, helper-functions, etc. already existed (as in, why were the cards ever in such a long switch statement if all the hard work had already been done?), so I simply used what already existed (i.e. drawCard, state, numActions, currentPlayer, etc.) to fulfill the requirements of each card. The infrastructure of the switch statement is so heavily relied upon that it would've required a significant overhaul of the codebase to extricate the functionality out completely, so each applicable case makes the function call.

And finally, once the card's had been encapsulated into their own functions, they were made available in the dominion.h file.

BUGS

You know it's tough to define a *meaningful bug*, not having much experience building in a bug (intentionally). For instance, building a test that includes checking for the expected behavior of a card make sense to me, but building a bug that randomly generates both expected and unexpected behavior seems malicious, but the difference in the test is no different; the card either behaves as expected or it doesn't and should direct the developer to the card for review. This all to say, the bugs I developed effect the behavior of the card in such a way that it does not perform as expected.

Specifically, I commented-out or tweaked values to alter things like the number of ACTION and/or BUY turns the card is supposed to dole out. The bugs allow for continued play, but deeply effect the outcome of the game, and thus, I would argue are meaningful and useful bugs.

GREAT HALL:	No bugs. Kept clean.
VILLAGE:	BUG: Provides four (4) ACTION turns rather than two (2)
COUNCIL ROOM:	BUG: Should provide another BUY turn, but does not
ADVENTURER:	BUG: Increments the drawn treasure by four (4) rather than one (1)
SMITHY:	BUG: Draws four (4) cards but should only draw three (3)