Code smell / Refactor 1:

Date Found: 12/2/23

Code smell: SRP Violation

- In TetrisStartMenu, TetrisPauseMenu and TetrisEndMenu the initialize functions were violating the SRP principle by calling the function components of the UI as well as customly creating the the background and then applying it.

Refactor: Extract Method

- To fix this SRP violation, a drawBackground function was created to minimize responsiblities of the intialize function.

Location in Source Tree of Refactor:

- TeamA-Tetris-Project > src > TetrisStartMenu.py

- TeamA-Tetris-Project > src > TetrisPauseMenu.py

- TeamA-Tetris-Project > src > TetrisEndMenu.py

Code smell / Refactor 2:

Date Found: 12/2/23

Code smell: Magic Numbers

- In tetris_verFinal.py the construction of the game buttons used many magic number not even hinting what they were for.

Refactor: Replace magic Number with Symbolic Variable

- This was fixed by declaring the variable first and then passing it to the call of the classes.

Location in Source Tree of Refactor:

- TeamA-Tetris-Project > tetris_verFinal.py

Code smell / Refactor 3:

Date Found: 12/2/23

Code smell: Duplicate Code

- There are 2 version of the Button class which creates the Duplicate code, code smell.

Refactor: Remove Duplicate Code

- This was refactored by removing one of the button classes and all calls to that class.

Location in Source Tree of Refactor:

- TeamA-Tetris-Project > src > Button.py

Code Smell / Refactor 4:

Date Found 12/2/23

Code smell: Dead Code

- Speed button class file has multiple unused methods and attributes

Refactor: Remove all dead unused methods/attributes in SpeedButton class

- This was refactored by removing all the unused methods in this class. These include the methods create_text_surface(), clickAction(), initiliaze(), and keyAction(). The attribute darkModeToggle was also removed

Location in Source Tree of Refactor:

- TeamA-Tetris-Project > src > SpeedButton.py

Code Smell / Refactor 5:

Date Found: 11/22/23

Code Smell: Duplicate Code

- Certain colors like BLACK and WHITE were being called from different sources and effectively being duplicated

Refactor: Remove duplication and have one main source of colors

- The colors are now all being pulled from the Theme() class for the most part. Duplicate instances of common colors have been removed

Location in Source Tree of Refactor:

- TeamA-Tetris-Project > tetris_verFinal.py

Code Smell / Refactor 6:

Date Found: 12/2/23

Code Smell: Speculative Generality in DarkModeButton

- Unused methods in DarkModeButton class

Refactor: Delete unused methods in DarkModeButton class

- Unused methods like keyAction() and generate_icon() which would have had different purposes in the future were deleted.

Location in Source Tree of Refactor:

- TeamA-Tetris-Project > src > DarkModeButton.py

Code Smell / Refactor 7:

Date Found: 12/2/23

Code Smell: Comments

- Many explanatory comments in block. Some of the comments in this file are also outdated

Refactor: Remove unnecessary comments & rewrite outdated comments

- Remove explanatory comments with no purpose and also rewrite the outdated comments. New comments were added in sections of the code that were also very confusing.

Location in Source Tree of Refactor:

- TeamA-Tetris-Project > tetris_verFinal.py

Code Smell / Refactor 8:

Date Found: 11/29/23

Code Smell: Heavy Coupling

- Heavy coupling in main() file between difficulty adjuster and keyboard pressing 1 & 2

Refactor: Removing unused code & changing access to fall speed modifier

- The code that allows the fall speed to change with the 1 & 2 keys has been adjusted to no longer affect the difficulty. This functionality has been removed.

Location in Source Tree of Refactor:

2 Locations (1.) TeamA-Tetris-Project > src > tetris_verFinal.py, (2.) TeamA-Tetris-Project > src > Difficulty.py

Code Smell 9

Date Found: 12/2/23

Code Smell: Long Parameter List

- SpeedButton has a very long parameter list. This is a code smell because it makes initializing/using this class very involved and difficulty. Some of these parameters are in the initialization of the button, but they also exist in methods like pygame.Rect(). There are up to 6 parameters being used in this __init__ of SpeedButton

Location in Source Tree of Refactor:

- TeamA-Tetris-Project > src > SpeedButton.py

Code Smell 10

Date Found: 12/2/23

Code Smell: If/Else

- getDifficulty() method has difficult to expand if/else chain for future feature additions. This is also heavily coupled in tetris_verFinal.py where these difficulty states in the if/else chain are used directly to change the button appearance. Any slight adjustment to either of these places could cause the difficulty functionality to break easily.

Location in Source Tree of Refactor:

- TeamA-Tetris-Project > src > Difficulty.py