

CMPT276 D100

Group 12

Cordell Bonnieux

Sikij Karki

## Assignment 4: Code Review

During the code review session, we together identified a series of bad code smells, which we took note of and addressed in the following weeks.

### Cordell's Changes

Problem	Solution	Commit ID
Application.java's state was contained as a private enumerator within Application; thus, state cannot be changed or checked from outside of the class.	Created public enumerator ApplicationState, and moved states out of Application into this new class.	6fd8420
In Game.java, the logic for freezing and unfreezing enemies was convoluted and needed to be simplified; it was prone to error.	Separated freeze & unfreeze logic and created an unfreeze timer to be called after enemies have been frozen, separating the two actions and eliminating the possibility of several related bugs.	3f557d3
Enemy.java's movement was tied to player actions, which worked alright but did not allow the player to use quick thinking to their advantage with swift movement; furthermore, when the player was trapped, the enemy ceased movement.	Decoupled enemy and player movement by creating a timer that then calls the enemy movement logic; furthermore, refactored and simplified the player's own movement logic for better readability.	587f407
Determining start positions for the player and enemies was being handled by	StartTile location finding was moved from Game to Map, and whilst doing so,	df3904d

Game.java requiring properties of Map.java to be public, which is an anti-pattern.	the algorithm was also improved.	
Classes that inherited Menu had repeated code blocks and, in addition, did not allow the flexibility of adding an image in lieu of a text title, which was part of our initial design.	Created an overloaded version of initializeUI for in Menu, which allows children to have their specified header images and eliminates repeated code.	e68e3e7

### Sikij's Changes

Problem	Solution	Commit ID
PlayerHUD.java used magic literals and hard coded formatting. Lots of repeated code, and unclear semantic meaning resulted in a difficult to maintain, modify and comprehend code	Named constants for previously hard-coded values, and format strings. Ensures all color usages and label initializations are aligned with the defined constants, promoting visual consistency in the HUD	9952342
Enemy.java's moveInRandomDirection() used a switch with lots of low readability code repetition. Combined with hardcoded magic numbers which make future maintainability worse.	Used an Enum for directions to eliminate the switch and extracted magic numbers into constants. Code is readable, and more extensible in further directions.	f93ee36
Cluttered codebase with warnings, unused code, and method signatures lacking proper type safety.	Remove unused variables, methods and imports. Enhance typesafety with proper generic use, and all code warnings throughout the entire codebase.	4fdefe0
Useless conditionals in Time.java for redundant code	Simplify conditionals in Time.java	199794b