Readme

Author: Dylan Corr
Module: Advance OO Software Development
Lecture: Dr John Healy.
Assignment: Refactoring and Redesigning a Video Game

Summary of refactoring with rationale.

- Created new class called GameUI. Separate the initialization and execution of the GUI from the main method. Rationale: The Runner should have a single responsibility SRP i.e launch the game.
- Take the model and objects arrays out of Runner class and put them in the GameUI class Rationale: They don't belong in the Runner class (SRP).
- Abstract the Direction Enum into a separate class, so that it can be reused independently of the Sprite class. Created an interface Direction, which the Directions Enum implements. Rationale: This will make it easier to add new directions i.e., open for extension. (OCP)
- Take the inner Point class out of the Game View class and make it into a separate class and change the code to use this new Point class. Rationale: The Point class doesn't belong in the Game View Class. Make it reusable. (SRP)
- Remove the image loading functional out of GameView class, create new class called Assets that stores the images read from disk. Rationale: The loading of files does not belong in the Game view class. (SRP)
- Remove the getIsoX(int x, int y), getIsoY(int x, int y) and getIso(int x, int y) methods from the GameView class. Using the strategy pattern, create interface CorTransform, and create new classes, IsoTransform and CartTransform, each of which implements this interface. Rationale: The methods replaced don't belong in the Game view class (SRP). The Strategy design pattern used which allows you to define a family of algorithms, encapsulate each algorithm in a separate class, and make the algorithms interchangeable at runtime. (SRP, LSP, OCP)
- Remove the method to change the colour of sprites, and put this method into another class, called SpriteHelper. Rationale doesn't belong in the Game View class. It breaks SRP.
- Change model and object arrays from type int to byte. Rational: using numbers that won't exceed 255.
- Created a Singleton GlobalVars provides global access to a set of variables. Rationale: Control over object creation and number of instances, prevent other classes from creating instances of it. Ensures that there is only one instance of the GlobalVars class in the application.
- Created a Factory class SpriteFactory. Rationale: Satisfies Single Responsibility SRP because it is only used to create sprite objects; follows the Liskov Substitution Principle (LSP) because it returns objects that all adhere to the same Sprite interface.