Project 6

CSCI 5448

DENNIS WINDHAM

NEILL SHIKADA

CONTENTS

Date: November 16, 2022.

STATUS SUMMARY

Title.

Object Oriented Civilization Game Clone

Team Members.

Name	Role	Email
Neill Shikada	Graduate Student	Neill.Shikada@colorado.edu
Dennis Windham	Graduate Student	dene5275@colorado.edu
Bruce Montgomery	Instructor	bruce.r.montgomery@colorado.edu

Work Done.

- **Neill Shikada**
  - Implemented complete AI that plays the game against other AI-controlled civilizations.
  - Created UI using Unity's systems.
  - Added audio support with royalty-free, copyright-free sound effects and music.
- **Dennis Windham**
  - Implemented board generation for the game.
  - Implemented unit, behaviors, unit types, civilization types.
  - Implement factories, graphics observer.

Changes of Issues Encountered. Unity generally advises tight coupling with their MonoBehavior base class, which is the only way to interact with the actual Unity object representations. We are avoiding MonoBehavior entirely for the underlying game logic, so we had to come up with a few workarounds when it comes to code interaction with Unity itself, e.g. a designated adapter-like helper class that does inherit from MonoBehavior. It was also initially unclear how we would facilitate graphics generation, but over a number of design iterations we decided to make use of the Observer pattern.

Patterns.

- Strategy
  - We make heavy use of the strategy pattern to dynamically assign behaviors to Unit objects.
- Factory
  - We have multiple factories in place to generate various game objects. Namely, factories create individual Civilizations, Units and all Graphics objects.
- Command
  - We are using the Command pattern to bridge UI buttons with in-game actions for the human user.
- Observer
  - The Observer pattern is being used to update the visual representation of the game board. Whenever an object with a graphical representation spawns or gets destroyed the respective graphics observer is notified and performs an update.
- Singleton
  - Singleton is being used for the persistent game settings object passed between scenes.
  - GraphicsObserver is also implemented as a singleton so as to enforce its role as the only object that interacts with Unity graphics.

Class Diagram.

See Appendix A: Demonstration Class Diagram for our class diagram.

Plan for Next Iteration.

We'll implement user controls, an actual turn-based system and endgame scoring. We expect to spend another 10-15 hours to complete the project.

Appendix A: Demonstration Class Diagram