SIGNIFICANT CHANGES FROM OBJECT-ORIENTED DESIGN

Functions void PromptUser(&screen); char UserInput(Menu &menu); bool ValidCommand(char input); Were removed from the Puzzle subclasses since they will be handled by the specific menu for each Class instead.

Due to the over-complicated nature of the builder pattern being used for the creation of characters, the design pattern will no longer be implemented. Instead there will be an abstract base class called Character.h that will have derived Npc.h and Player.h classes. The derived classes will have hard coded information for the specific character object being created.

WORK DISTRIBUTION

Screen/ File I/O/ Image- Reid will code and test Screen, File I/O. Tyler will help testing Puzzle related screen and File I/O functionality. Including: Screen.cp Screen.hh SlotScreen.cp SlotScreen.hh Room.hh DefaultImg.hh Image.hh ImageImporter.hh ImportImg.hh

RoomTree/Item- Rylan will code and test RoomTree Item and all subclasses of Item. Including: RoomTree.h Bow.h Consumable.h Item.h MyConsumables.h MyWeapons.h Spell.h Weapon.h

Menu/GameState- Tomas will code and test Menu's and their subclasses and will code GameState and test it. Tyler will also test the puzzle related derived classes of Menu,

Characters- Gates will code test Character.h class and its derived classes Npc.h and Player.h.

Puzzles- Tyler will code and test Puzzle, and its subclasses. Including: Puzzle.h/.cpp, CodeCracker.h/.cpp, Hanoi.h/.cpp, MemoryMatch.h/.cpp, and TicTacToe.h/.cpp.

Main- When the first person finishes implementing/testing their specific classes they will begin work on the main program, as others finish they will contribute to error handling/testing in the main.