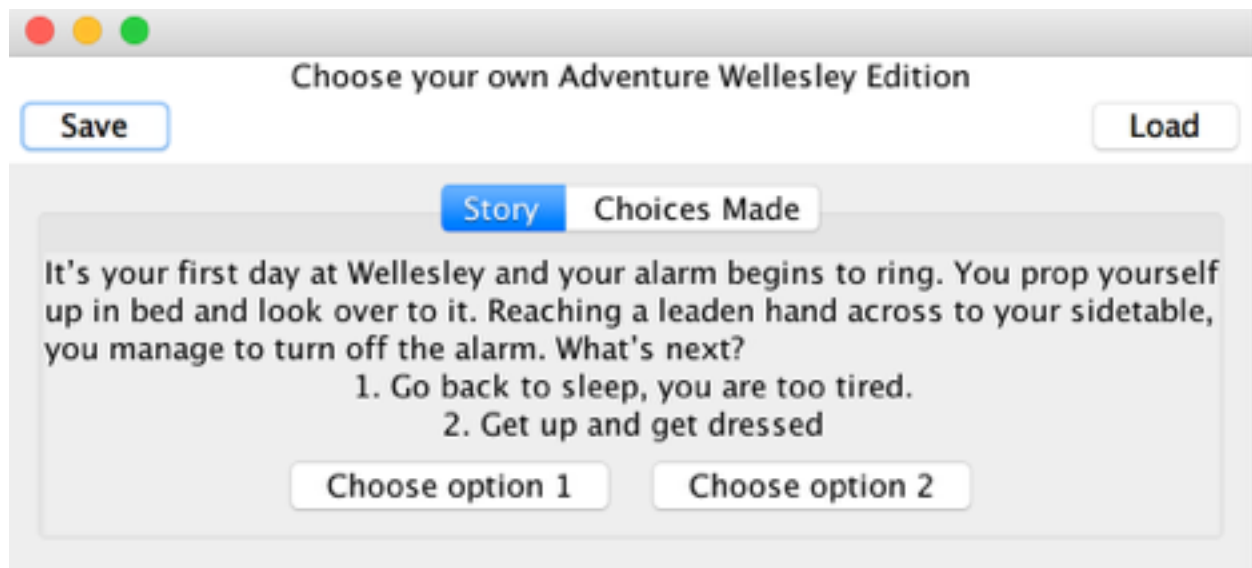


User Manual:



Shows the game's UI, players click on the option 1 or option 2 buttons to choose which decision to make down the tree. Save and Load buttons will allow the user to go back to their previous choices and retry them. Choices made reminds the user of what choices they took to get where they are. They also persist when the load function is called, meaning that the player can remind themselves what they did if they load from a save made in a different sitting.

Classes:

GameGUI:

Acts as driver class, contains all relevant GUI elements for the frame.

METHODS:

- GameGUI: constructor
- makeNorthPanel: returns a JPanel with the information about the game and the save/load buttons
- ButtonListener subclass: creates custom ButtonListeners for the GUI
- Main: holds official game code, run the file to start the game

Player:

Contains information about the Player object. Holds "choices" the player makes in a LinkedList that is treated as a Queue. Contains save/load functions.

METHODS:

- Player: constructor
- save: dequeues from the linked list the choices the player made in traversing the tree and writes them to a txt file.
- load: traverses the tree based on the inputs taken from the designated text file

StoryPanel:

Makes the JTabbedPane that contains both the story itself, the choices, and the player's choice history.

METHODS:

- StoryPanel: constructor
- updateStoryPanel: updates the labels and buttons on the story panel to show the current state of the tree
- choicesPanel: returns a JPanel that contains the list of choices the player made
- getOpt1/getOpt2/getGrabText/getChoices/getChoose1/getChoose2: getters for each element in the panel so as to be accessible through other files
- ButtonListener class: creates a specific button listener for the two choice buttons, which traverse the tree left or right depending on the button pressed

StoryTree:

Creates binary tree for StoryNode objects by reading from a txt file.

METHODS:

- StoryTree: constructor
- showCurrent: getter for the long story text of the current node
- getCurrentIndex/getCurrentNode/getOpt1/getOpt2/getShortString/: getters for different elements of the node
- advance: given an input(left or right) traverses the tree accordingly.
- toString: returns a string representation of the current node
- startOver: resets the story to its base state, with currentNode set to spot 0 in the Vector that contains the nodes and currentIndex set to 0
- isEnd: tests to see if currentNode is a leaf
- main: testing for the tree

StoryTreeNode:

Base holder for information needed for the story

METHODS:

- StoryTreeNode: constructor
- StoryTreeNode: alt constructor, takes in a string
- getShortText/setShortText/getOpt1/setOpt1/getOpt2/setOpt2: getters and setters for different fields
- toString: returns the node's content i.e. the story string
- visit: sets the visited boolean to true, not used but left in for posterity

Common Field Names:

Content: Long text for the story, shows up in main story panel as the text above the two options

Opt1: the first of two options, sits above opt2, goes to left child node

Opt2: second of two options, sits below opt1, goes to right child node

ShortString: shortened representation of the option that the player chose to get to this current node i.e. short representation of the parent node.