**Instance Variables and Methods**

class Editor:

* **Int CursorX**
* **Int CursorY**
* **int LINE\_LENGTH**
* **private FastLinkedList buffer**
* **void delete()**
* **void add(char x)**
* **public void handle**
* **public String textToString(double startX, double startY, double endX, double endY)**
* **public String stringToString(double startX, double startY, double endX, double endY)**

class KeyEventHandler

class MouseEventHandler

class ScrollBarHandler

class Print:

* **public static void print(String toPrint)**

class Text: javafx.scene.text.Text

class History

* **Stack<HistoryEvent> undo**
* **Stack<HistoryEvent> redo**
* **Public void newEvent()//push to undo and clear redo**
* **Public HistoryEvent operateUndo() //pop from undo, push to redo and return**
* **Public HistoryEvent operateRedo() //peek from redo**

class HistoryEvent:

* **private KeyCode code**
* **private Text text**
* **private int x**
* **private int y**
* **get()\*4**
* **set()\*4**

class Cursor:

* **Rectangle textbox**
* **Int curX**
* **Int curY**
* **Public void setX(int x)**
* **Public void setY(int y)**
* **Public int getX()**
* **Public int getY()**

class FastLinkedList

* **private Node sentinel:** Sentinel node for the linked list.
* **private int currentPos:** Character number in the text.
* **private Node currentNode:** Current node to be inserted after.
* void addChar(char x): This adds the given character to the linked list after the currentNode.
* void deleteChar(): Deletes the currentNode.
* int currentPos(): returns the currentPos
* int getCurrentPos()
* int setCurrentPos()
* class Node
  + Text nodeText: The character for the current node.
  + Node prev, next:

Runtime constraints:

* Inserting/deleting characters: Must be constant time.
* Changing cursor position with arrow keys: Must be constant time.
* Changing cursor position with clicks: Must be constant time.
* Rendering text: Must be linear time.

**Analysis: Insertion/Deletion**

* Insertion/deletion: Since we have a pointer to the current node, insertion and deletion is very fast.
  + We are not counting the time needed to update all the x and y coordinates of the Text objects in the linked list. This is fine, because this is part of rendering.

**Analysis: Changing Cursor Position With Arrow Keys:**

* User pressed down arrow: We can scan forwards, using JavaFX methods to check width of each letter as we go. Resulting runtime is constant (since it’s just the number of characters to the next line).

**Analysis: Changing Cursor Position With Clicks:**

* Scanning forwards like in the arrow keys example will take linear time in the worst case, since the user might click at the end of the file! This means our data structure is not sufficient!

**Analysis: Rendering Text**

* [Not discussed].