Test scenarios/user tests/ stress testing:

**Test Scenarios:**

* Creating and saving a file, switching off computer (or closing the program), then trying to access it again

Bobert has created a file in pick-a-path and once he finished working on it at the time, he saved his file and closed the program. Later, he decided that he wanted to add a few more scenarios and decisions, so he re-opened pick-a-path, opened his save file, and added more boxes and arrows. He is satisfied with his work, and saves his file and closes the program, then later opens it and has his friend, Fred, play through it. After getting feedback from Fred about his game, Bobert decides to add even more scenarios and decisions, so he opens pick-a-path again and adds more boxes and arrows, then saves the game again and exits the program.

* Power cuts while using program

Bobert has been a working on a program for an hour. He saves the program. But after a couple of minutes he starts to make changes in the program again when abruptly his laptop runs out of battery and switches off. He charges his laptop, restarts the application and opens his saved file from the computer. He finds that his most recent changes that were unsaved are gone. The file only consists of the data from the last save point.

**User Tests:**

Jimmy: having his sister try to use the program pros and cons

Result: His sister found it easy to use, though the “add arrow” part was a little confusing until he explained it to her. Other than that, she thought it was easy and it was fun to create a story in a “choose your own adventure” format. After looking at the manual/having jimmy explain things, she was able to use it as we do when we make test stories.

Logan: having his roommate use his auto clicker to create A LOT of boxes and seeing if it crashes

Result: boxes kept populating, and this did not slow down the program or crash it

**Stress Tests:**

* (Logan’s roomie’s auto clicking) adding a bunch of boxes, seeing if that crashes upon creation or when we play the story (seeing if the program has slowed down or crashes): this did not crash or slow down the program, it appeared to be creating boxes at a constant rate until Logan stopped the auto clicker.
* Manually adding a lot of boxes (connected) and seeing if there’s a long story with a lot of options that will break the program: the program doesn’t crash on account of the number of boxes and arrows. You can add as many boxes and arrows that you want. But these boxes and arrows need to contain text (names/numbers) of some sort in them in order for the program to work. Empty boxes and arrows can be saved in a .pap file but will crash in the player mode.
* Using multiple windows of the program at the same time: doesn’t crash, you can work on the same project in multiple windows and update them as you want, the most recent save is the most recent save for the file