

Bridging the Gap: Storytelling Alice as a Precursor to Python

Amanda Gilbert

Computer Systems Lab 2009-2010

Development and Discussion

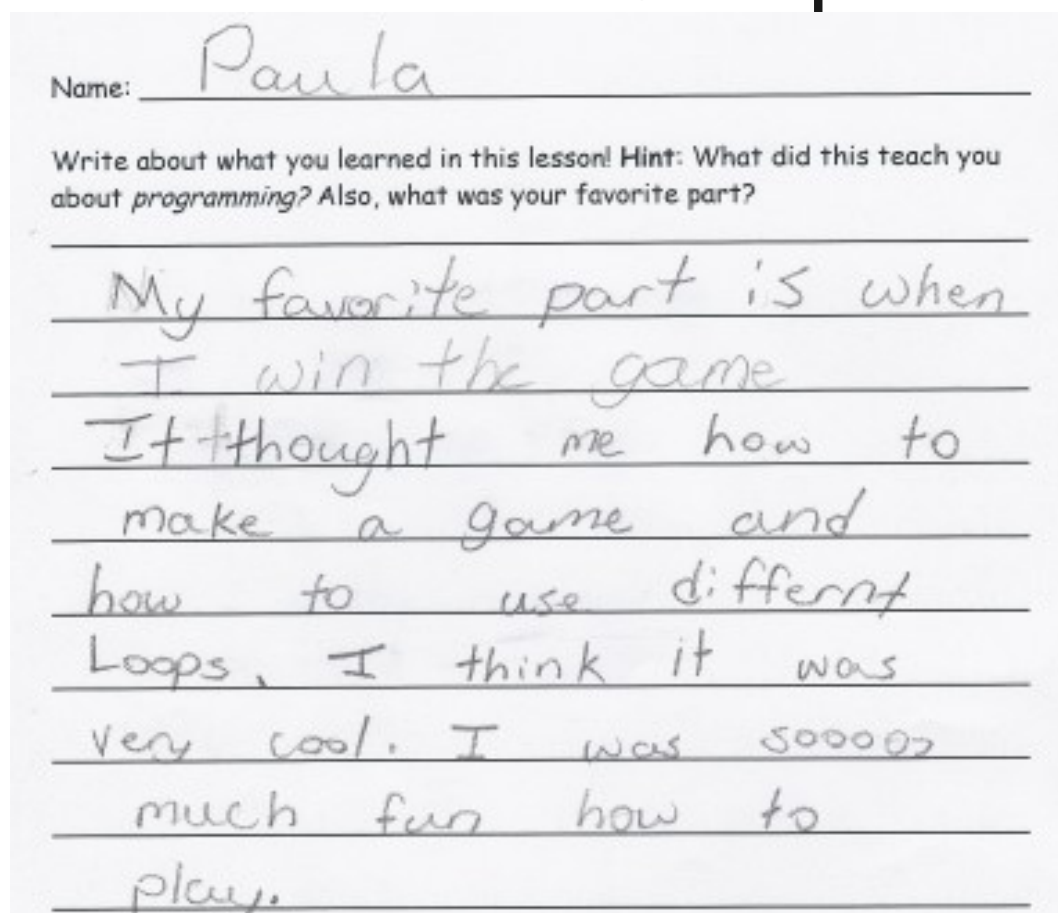


Figure 1

Abstract

The goal of my project is to determine whether or not Storytelling Alice is successful in preparing elementary school students for a career in computer programming. Essentially, I am testing to see how possible it is to bridge a gap between a visual and kid-friendly interface like Storytelling Alice and a non-visual, syntactically difficult interface like Python. The students ability to program in Python will be indicative of Storytelling Alice's success at preparing students for a career in programming.

Background and Introduction

Storytelling Alice was created by a student at Carnegie Mellon University named Caitlin Kelleher. Last year, two girls did a research project at Cardinal Forest Elementary but focused on Scratch Programming. I was put in charge of teaching new languages, Storytelling Alice and Python to a fourth grade class. In her PhD dissertation, Kelleher recognized how important diversity is in the field of computer science. She felt Storytelling Alice could help attract middle school girls to computer science and geared her language toward that group. A study found that boys' and girls' ideal technologies were significantly different, meaning that having a more equal ratio of girls and boys in the field could drastically change the course of technological development.

One article stated that girls have the ability to program as boys do.

There are many reasons why the field is male dominated: there are social norms that encourage boys to program and not girls, and at the middle-school level, girls' confidence in their abilities relating to math and science becomes deflated. This is actually a very promising fact for my study. Because we are working with Elementary School students, not only are we able to encourage programming for girls before they reach middle school level and lose confidence, we are also working with them at an age where they are unlikely to know about social norms.

There are other benefits to using Storytelling Alice. Concepts that are hard for first-time programmers to understand like variables, parameters, and loops have to be introduced to assignments in other languages. Many problems like invisible state and syntax errors are not an issue in Alice.

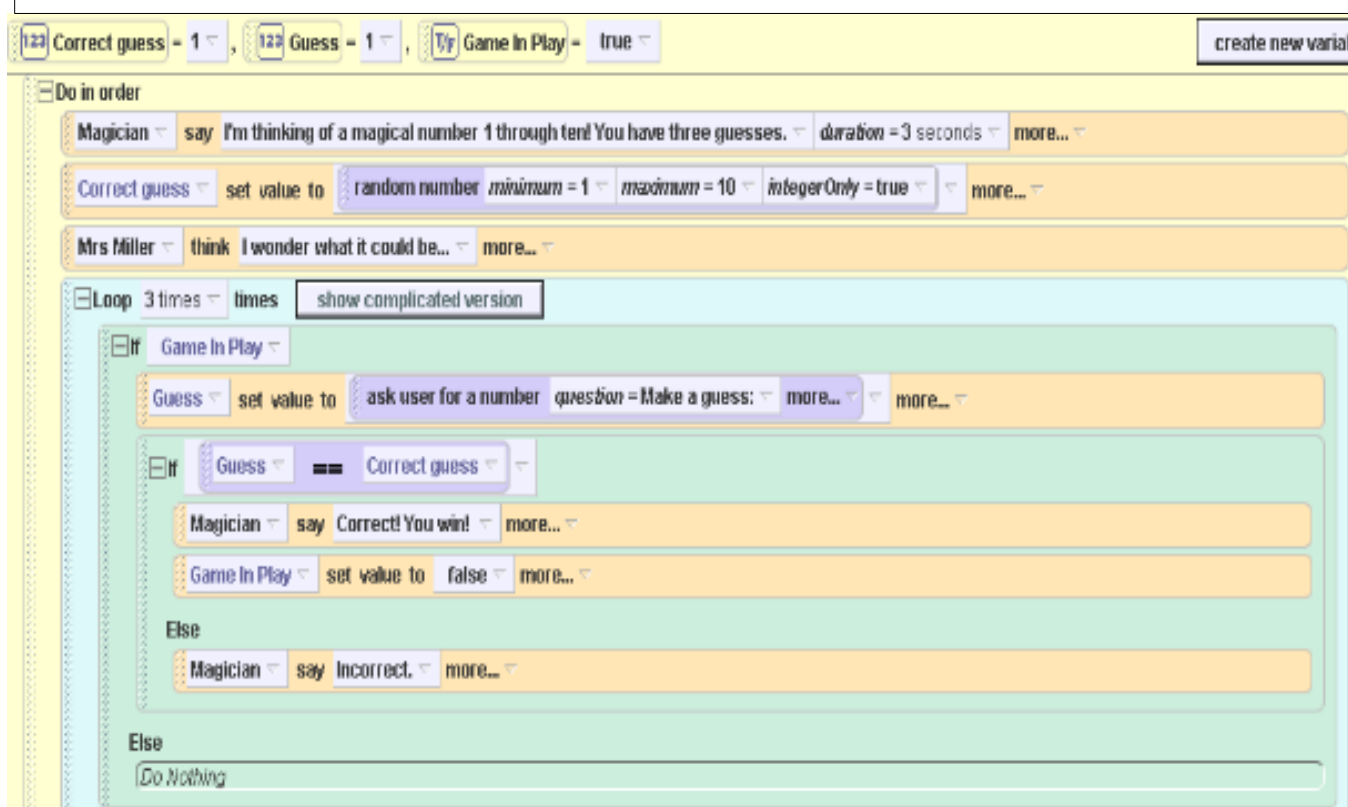


Figure 2

Each Tuesday, I work with a class of fourth graders. There are only three boys in the class (the majority of the students are female). We completed four projects in Storytelling Alice this year and two in Python. The first was an introduction to the Alice interface. It taught the kids how to create a world in Alice and allowed them to get comfortable with the new environment. The second lesson focused on creating methods and the third on creating stories in Alice. Each lesson targeted to teach certain programming concepts and over the year so far the students have learned about different control statements, variables and methods. The final program in Storytelling Alice was called "Guess that Number". It centered less on the visual aspect of Alice and more on the coding. This lesson was also the first project we looked at in Python, so that the concepts were the same but the language and environment were the only differences. I also have changed my methods of assessment over the course of this year. After we introduced Python with the familiar project, Guess that Number, we made modifications to this code together in class to create an extension of this project with modifications to the code. I gave three quizzes and allowed the kids to write a journal and complete a worksheet. See results and conclusions for the data acquired. I found that many students are threatened by the concept of the quiz and I am able to get more accurate data through other means of assessment.

Results and Conclusions

I have collected five sets of data from different formal assessment methods throughout the year, while also collecting qualitative data, such as my observations and my conclusions after studying the students' completed projects. On the first quiz, there were five perfect papers out of sixteen. This was a promising start to the year. However, the second quiz asked the students to code on their own without prompting and without the help of the drag-and-drop interface. Many were too intimidated by this concept to do well on the quiz and therefore the results were not good- the average score was about a 48%. From this I learned that asking students to code individually before they gained confidence in programming would yield unsuccessful results, and therefore I must be careful in deciding when to make the transition. This also caused me to rethink my methods of assessment and after the fourth lesson I decided to give the kids a journal assignment where they could write about what they thought about the lesson and what it taught them. One example of a student's response is found in figure 1. I found that I was able to understand how the children felt and how confident they were with coding without the prompting or intimidation that comes with questioning. However, the downside to this method is the inability to look at this data in terms of numbers to assess the students' understanding as a whole.

We then did a comments worksheet in class that served as both a learning and assessment tool. The results of this were more promising than the previous quiz. The average score was an 88%. The final quiz was a multiple choice, accumulative exam and all students did very well-- on every question but one, the majority of the class answered correctly. I knew from these results that my students were adequately prepared in Storytelling Alice to advance to Python. Figure 2 shows the code Guess that Number in Storytelling Alice and Figure 3 shows the same code in Python. After explaining the concepts in Storytelling Alice and going over basic differences between the languages, the students were each able to complete Guess that Number in Python (after being given a shell). I believe this is proof that Storytelling Alice is an effective precursor to Python programming.

```
from random import *
def main():
    print "I'm thinking of a magical number 1 through 10! You have three guesses."
    correctguess= randint(1,10)
    gameinplay=True
    for x in range(1, 4):
        if(gameinplay==True):
            guess= raw_input('Take A Guess! \n ')
            guess= (int)(guess)
            if(guess== correctguess):
                print "Correct! You Win!"
                gameinplay=False
            else:
                print "Incorrect."
if __name__=='__main__':
    main()
```

Figure 3