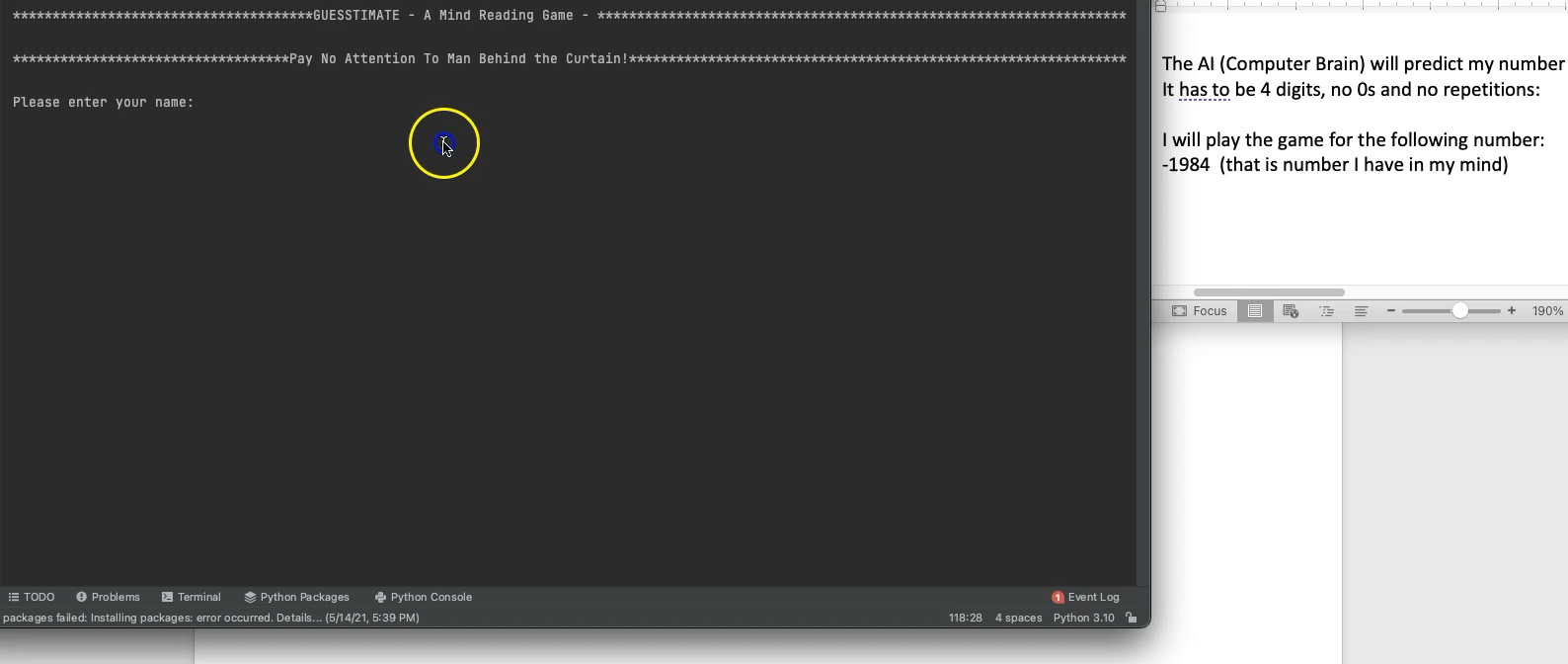
**Program: Guesstimate**

It is a game where the user thinks of a number then the computer tries to guess the number and the user provides feedback and after a certain amount of guesses the computer should find the right number.

This is a game where player picks a 4 digit number with no repeating numbers or 0s in its digits and computer predict that number via series of guesses.

(1984 is acceptable but 2010 is not b/c of 0s nor 1997 b.c of 2 9s.)

Each time computer makes a guess, user enters a feedback in format [+x-y]

(x is no of hits in right digit and y is no of hits in wrong digit)

For example, if the number is 1234 and computer's guess is 3264

1234

3264

2 and 4 correct digits, 3 in wrong digit. user will enter: [+2-1]

Double Click the video below to see the Game in Action!

**How it works:**

the user picks a 4 digit number that can’t repeat digits or have a zero value, and then the computer starts making guesses and the user gives feedback or “hints” based on how hot or cold they are to the number using a plus when a digit was guessed in the correct place value, and minus when the digit is there but in a different place value. For example, if the users number is 3572 and the computer guesses 3296 the feedback will be +1/-1. Plus one for the three, minus one for the 2. And this process continues until the computer guesses the right number.

**Main Algorithm**

1. User picks a 4 digit number that the computer will try to guess

2. Computer makes a guess

3. User inputs a feedback for the computer to show how accurate the guess was

4. Computer makes another guess based on the feedback that they got on step # 3

5. This process (step 2-4) continues until the computer guesses the right number. (feedback is +4)

**Sub Algorithm 1: CreateFeedback()**

1. check if the user guess is a valid entry (4 digits no zeroes and no repeated digits)

2. check the computers numbers thousands place and record if any numbers from the users thousands place input match

3. check the hundreds place and record if any numbers from the users hundreds place input match

4. check the tens place and record if any numbers from the users tens place input match

5. check the ones place and record if any numbers from the ones hundreds place input match

6. require user to input how close the computer was with the plus minus method

7. Repeat until number is found

**Sub Algorithm 2: CheckNumber()**

1. checks if the number has 4 digits

2. checks if any digit is a zero (only 1-9 is acceptable)

3. check if any digit is repeated

4. prompt for a new number if any of 1-3 is true

**List of Abstractions**

* guessANumber: Picks a random number inbetween the paremeters given
* repeatDigit: Checks to see if any numbers in the guess are repeated
* createList: makes a list to narrow down the possible right answer
* checkFeedback: checks the users feedback to see if it is written correctly
* processFeedback: converts the users feedback to code which the computer can understand