Approach to Problem Solving

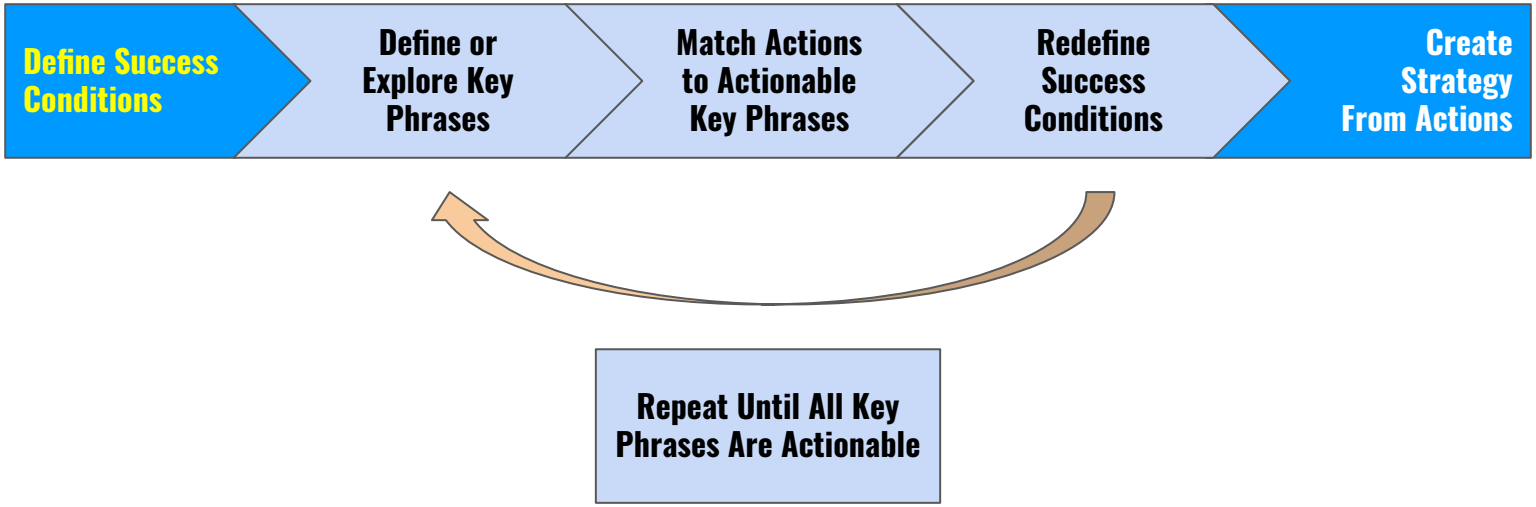
...

Objectives

In this lecture, we'll explore steps that are considered *an* approach to solving problems.

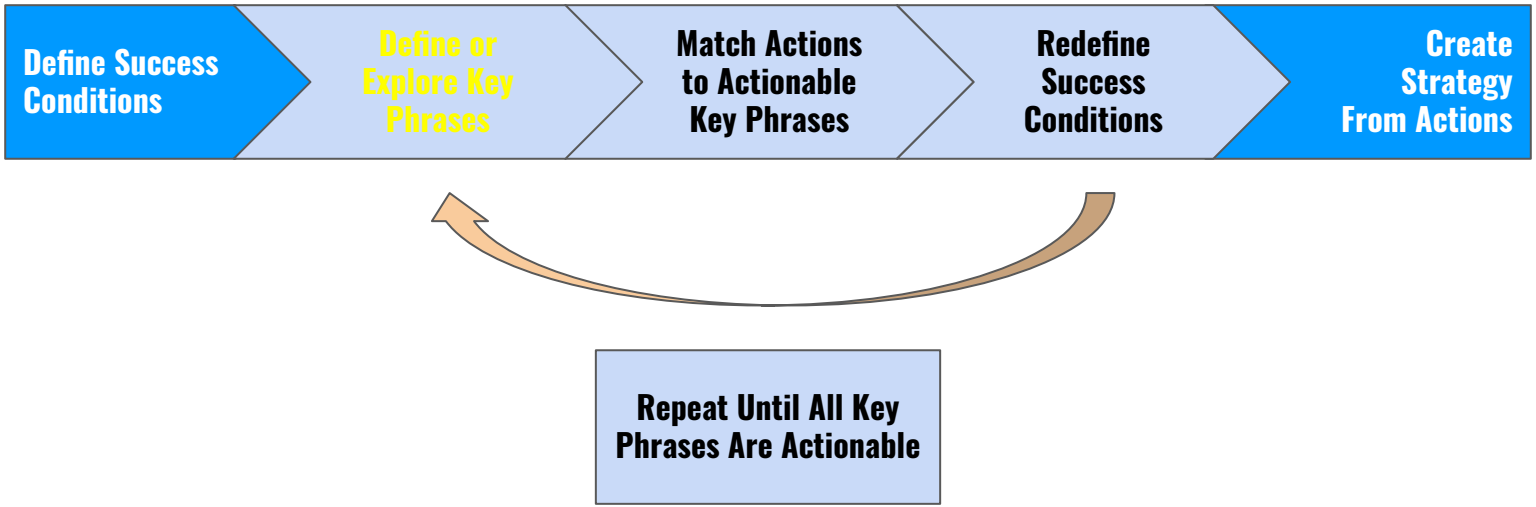
Remember, there are many different ways to solve a problem and this is just one of them :)

An Approach to Problem-Solving



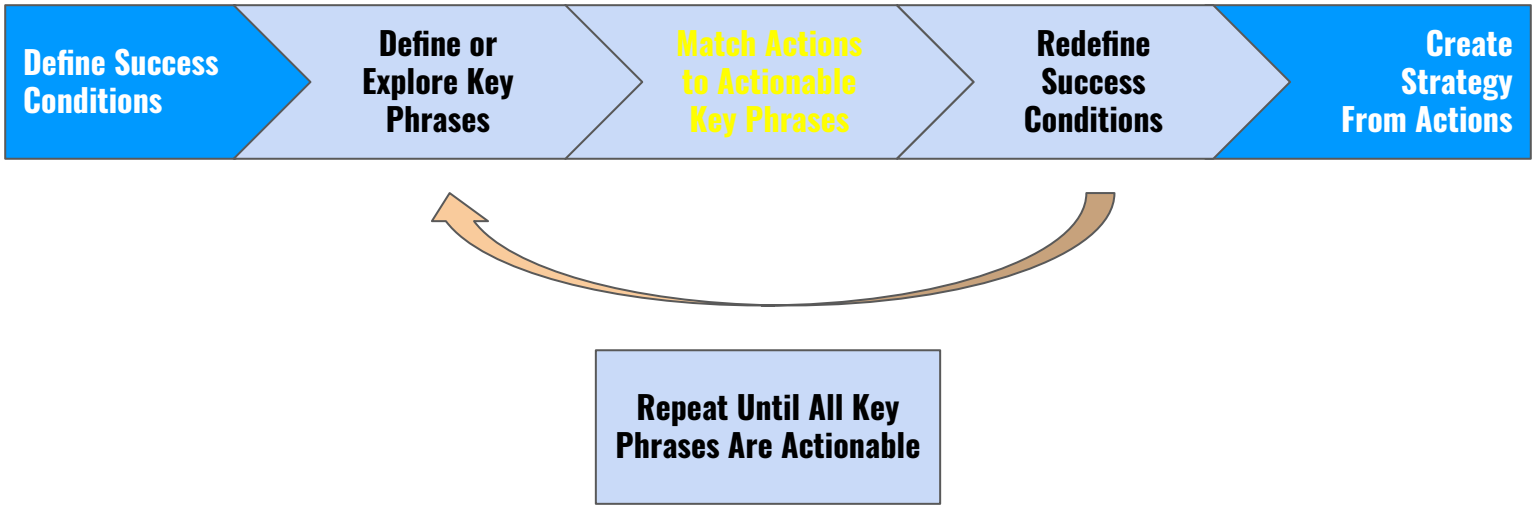
Define Success Conditions: *What about the question/prompt determines success? Ex. should something specific be returned? Will the execution have some sort of side effect? What are your outputs!*

An Approach to Problem-Solving



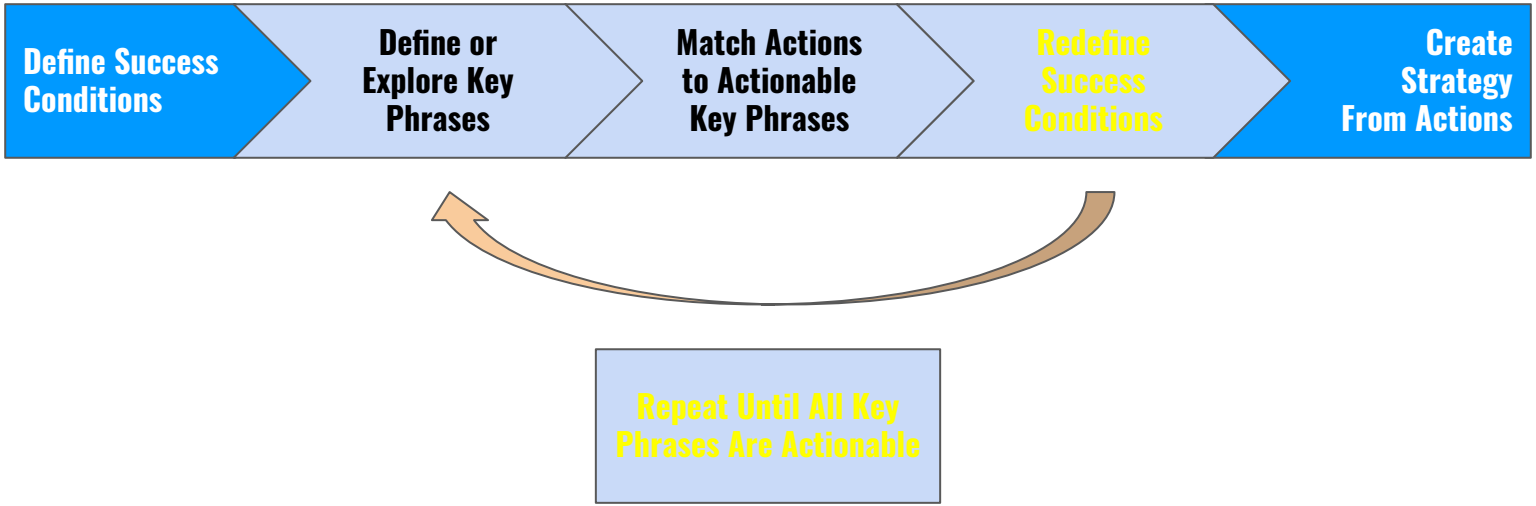
Define or Explore Key Phrases: *What information is provided? Input/Output? Type of data? Conditions about input/output? What information is available outside of the specified output that will help you get closer to success conditions*

An Approach to Problem-Solving



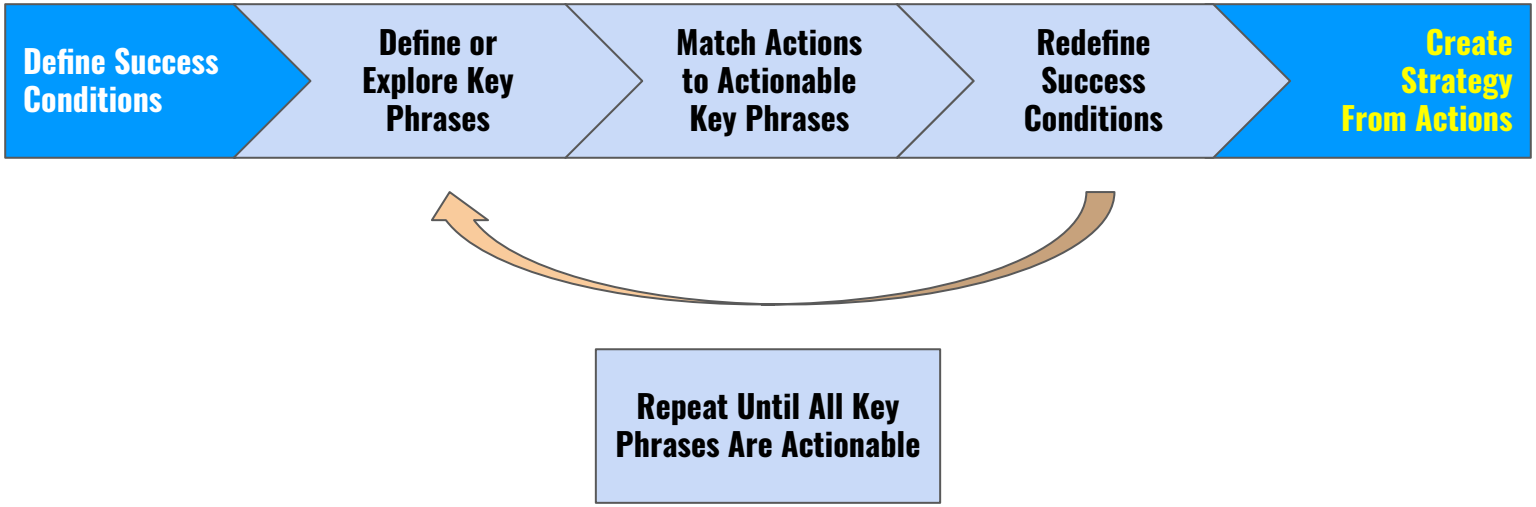
Match Actions to Actionable Key Phrases: *What can we do using code having these key phrases? For example, if we know we're receiving an array should we use iteration/looping? If we need to track values should we leverage a particular data structure?*

An Approach to Problem-Solving



Redefine Success Conditions: *Based on the provided information and what we've extrapolated from it, is there anything about our success conditions that we should reconsider?*
Repeat Until All Key Phrases Are Actionable: *If so, repeat process..*

An Approach to Problem-Solving



Create Strategy From Actions: *Take what information you've gathered and come up with a strategy to execute your code!*

Problem: Greed is Good

Problem: Greed is Good

Greed is a dice game played with five six-sided dice. Your mission, should you choose to accept it, is to return the score of a throw according to these rules. You will always be given an array with five six-sided dice values.

Three 1's	...	1000 points
Three 6's	...	600 points
Three 5's	...	500 points
Three 4's	...	400 points
Three 3's	...	300 points
Three 2's	...	200 points
one 1	...	100 points
one 5	...	50 points

Scoring Rules

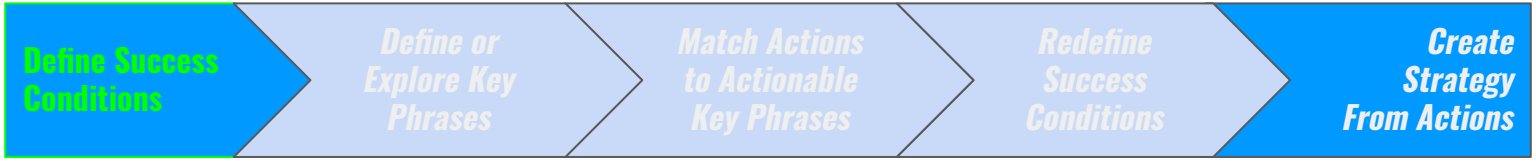
A single die can only be counted once in each roll. For example, a "5" can only count as part of a triplet (contributing to the 500 points) or as a single 50 points, but not both in the same roll.

What is/are our success condition(s)?

Greed is a dice game played with five six-sided die. Your mission, should you choose to accept it, is to return the score of a throw according to these rules. You will always be given an array with five six-dice values.

Three 1's	...	1000 points
Three 6's	...	600 points
Three 5's	...	500 points
Three 4's	...	400 points
Three 3's	...	300 points
Three 2's	...	200 points
one 1	...	100 points
one 5	...	50 points

A single die can only be counted once in each roll. For example, a "5" can only count as part of a triplet (contributing to the 500 points) or as a single 50 points, but not both in the same roll.

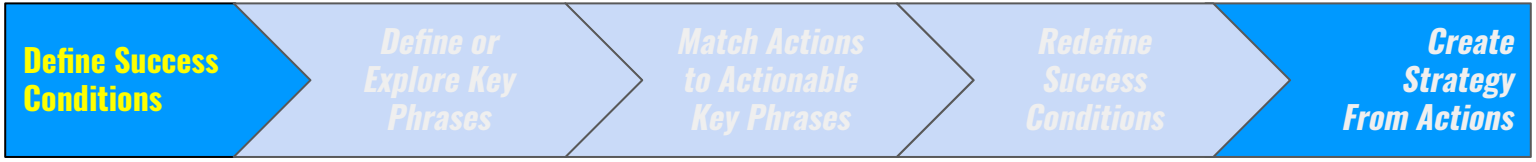


What is/are our success condition(s)?

Greed is a dice game played with five six-sided dice. ***Your mission, should you choose to accept it, is to return the score of a throw according to these rules.*** You will always be given an array with five six-dice values.

Three 1's	...	1000 points
Three 6's	...	600 points
Three 5's	...	500 points
Three 4's	...	400 points
Three 3's	...	300 points
Three 2's	...	200 points
one 1	...	100 points
one 5	...	50 points

A single die can only be counted once in each roll. For example, a "5" can only count as part of a triplet (contributing to the 500 points) or as a single 50 points, but not both in the same roll.

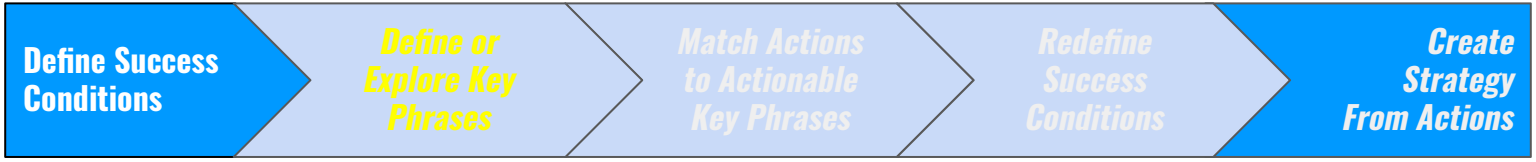


Key things (phrases) to keep in mind

Greed is a dice game played with five six-sided dice. **Your mission, should you choose to accept it, is to return the score of a throw according to these rules.** You will always be given an array with five six-dice values.

Three 1's	...	1000 points
Three 6's	...	600 points
Three 5's	...	500 points
Three 4's	...	400 points
Three 3's	...	300 points
Three 2's	...	200 points
one 1	...	100 points
one 5	...	50 points

A single die can only be counted once in each roll. For example, a "5" can only count as part of a triplet (contributing to the 500 points) or as a single 50 points, but not both in the same roll.

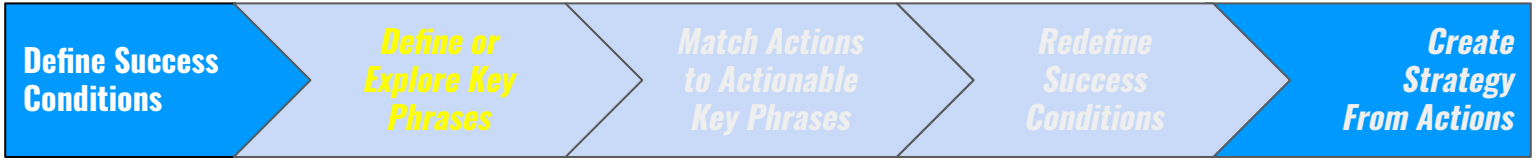


Key things (phrases) to keep in mind

Greed is a dice game played with five six-sided dice. **Your mission, should you choose to accept it, is to return the score of a throw according to these rules.** ***You will always be given an array with five six-dice values.***

Three 1's	...	1000 points
Three 6's	...	600 points
Three 5's	...	500 points
Three 4's	...	400 points
Three 3's	...	300 points
Three 2's	...	200 points
one 1	...	100 points
one 5	...	50 points

A single die can only be counted once in each roll. For example, a "5" can only count as part of a triplet (contributing to the 500 points) or as a single 50 points, but not both in the same roll.



Based on the info, what are some actions we can pull?

Greed is a dice game played with five six-sided dice. Your mission, should you choose to accept it, is to score a throw according to these rules. You will always be given an array with five six-dice values.

A single die can only be counted once in each roll. For example, a “5” can only count as part of a triplet (contributing to the 500 points) or as a single 50 points, but not both in the same roll.

So what should we do?

- Need to keep track of how many of each number came up in the throw. **How ?**

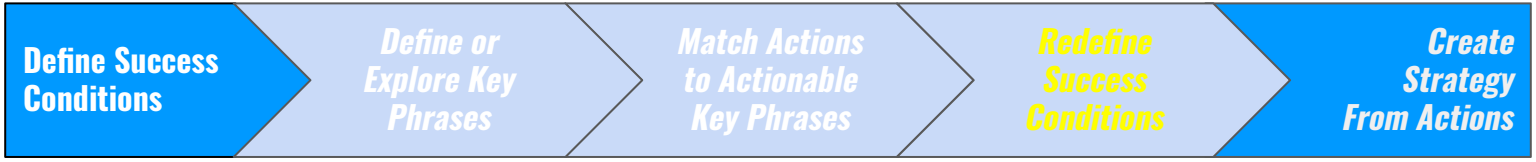
  [insert awkward group silence as I wait for suggestions]

- To get best possible score, we must first handle a triplet then handle singles. **How?**

Does this new information change our success conditions?

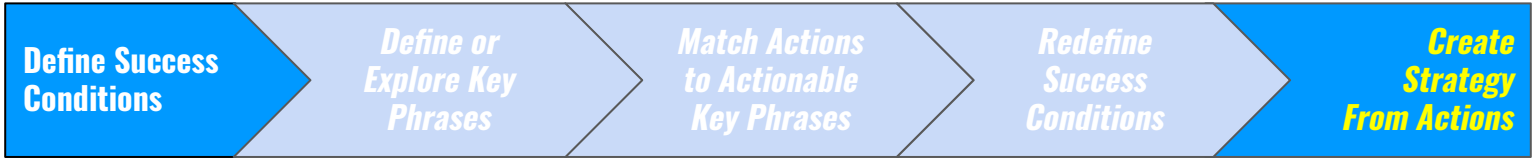
Success Conditions:

- Return value: Our output should be a number representing the score for our throw
- Considerations: Our best possible score can be calculated by dealing with triplet before singles, handling input in a way that does not count a single dice more than once



Strategy Time

Now we can create an actual strategy!



Building a Strategy..

1) Define input and output
2) Interpret input and output as JavaScript
3) Play around, consider example inputs
4) Describe approach using plain english *(or whatever you're most comfortable with!)*
5) Write pseudo-code
6) Write some code

Step 1: Define Input and Output

What's given to you?

Input: an array of 5 six-sided dice

What's are you giving back?

Output: the score of the throw

additional useful information?

- Scoring table
- Scoring rules (can only use one dice roll at a time)

Step 2: Interpret Input and Output as JavaScript

What's given to you?

Input: dice rolls

- **an array of numbers**, each representing a single throw of 5 dice (*greed[1, 4, 5, 4, 4]*)
- Dice only have 6 sides so our number will never be < 1 or > 6

What's are you giving back?

Output: the score of the throw

- a **number** representing the best total score of a given dice roll

additional useful information?

- Scoring table
  - **an object** containing a map of scores
- Scoring rules (can only use one dice roll at a time)
  - **if statements** representing score rules

Step 3: Consider the simplest examples (play around)

6, 6, 4, 4, 3
- Bad throw, what should we return?

1, 5, 6, 4, 3
- Looks like we only have 2 qualifying single numbers, so we can sum their corresponding point values and return that

1, **1, 1, 1,** 3
- We've got a triplet! … and more. How should we proceed?

Step 4: Describe approach to simplest cases in english *(or w/e you're comfy with)*

Example: Loop through input array keeping track of count for die number and updating score if a triple is found

Step 5: Create Pseudo-Code

Identify nouns with JavaScript data types, verbs with functions, and equivalence with assignment

- **Loop through input array keeping track of count for die number and updating score if a triple is found**

Becomes

- **Create triplet score by reducing the input data and caching each die count until we hit a triple and/or we complete the loop**

Step 5: Create Pseudo-Code

- Declare a variable ‘SCORES’ and assign it to an object containing property/value pairs that correspond to the given scoring system
- Reduce input data to determine score for triplets and get a count of any additional singles
  - While reducing, update a cache of die numbers, reset a die count if a triple is found so it is not additionally counted as a single
- Reduce remaining singles, initialized with score for triplets to determine final total with singles points (if applicable)
- Return final total

main benefit, we don’t think about code and our algorithm at the same time

Step 6: Write some code...

Step 7: Test

Step 8: Refactor