Lesson 1.2: Guessing Game Part 2

Objectives

In this lesson, students will:

- ❖ Recognize patterns and group data points by pattern
- ❖ Use abstraction to ask the correct question
- ❖ Using pattern recognition to make a better decision
- ❖ Create a decision tree and compare the efficacy of 2 different decision trees

Preparation

- ❏ Projector and speakers for video
- ❏ Print student activity worksheet (one per student or student pair)

Agenda

1. Introduction	10 mins
2. Decision Tree	20 mins
3. Making a better Algorithm	15 mins
4. Wrap Up and Reflections	10 mins

Resources & Links

- ❏ Shapes printed and cut (from the previous lesson)
- ❏ Video: Abstraction Computational Thinking

Credits:

Carnegie Mellon Robotics Academy

1. Introduction

Remind the students about the guessing games. What was the terminology used to group shapes into groups? (patterns)

The ability to recognize patterns is called “**Pattern recognition**”. When solving a problem, being able to find patterns and differences among things called data points is very important. Today, we are going to build a decision tree called an “**Algorithm**”.

Last time, we asked Yes/No questions to guess a shape. Let’s suppose I have 10 shapes I can guess from. Each shape has a number from 1 to 10. 5 shapes are green and 5 shapes are Yellow. Which question should I ask first:

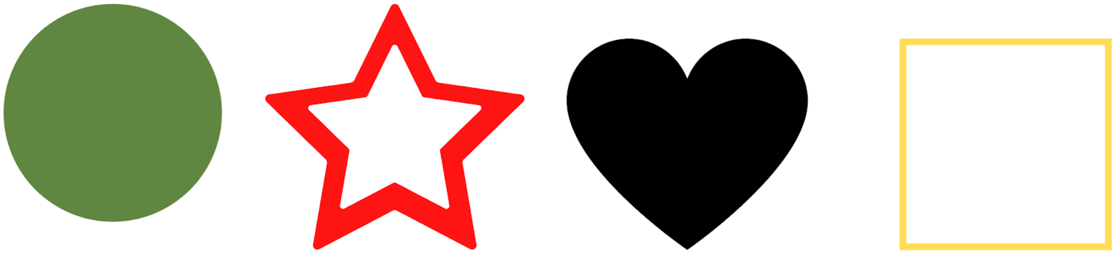
- Is your shape, shape # 1?
- Is your shape green?

Ask students for their opinion and open the debate. What if the shape was indeed shape #1? But it is likely?

An **Algorithm** is a set of instructions that can be followed by a machine called a computer, in order to solve a problem. The problem we are trying to solve is “Finding the shape chosen by another student among a set of shapes that we can see.” The algorithm is going to be a series of Yes/No questions that we need to ask in a certain order. The key is that our algorithm should work in the most optimal way, and because computers are not lucky people, the best algorithm is the one that will try to lead us to the least amount of questions.

2. Decision tree

Show the students the following shapes



Ask students what are the Yes/No questions I could ask to find a shape chosen by someone?

Collect all the answers.

- Is the shape red?
- Is the shape yellow?
- Is the shape black?
- Is the shape green?
- Is the shape full or outlined?
- Is the shape a star?
- etc..

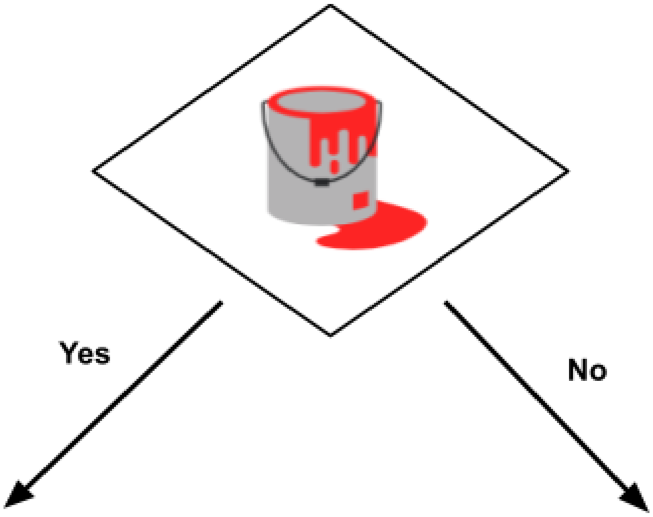
All of these questions are good because they help us either find the mysterious shape or at least eliminate one shape for the lot. But is one question better than another? To find out, we are going to build a decision tree.

A decision tree starts with a Yes/No question and has 2 arrows below: one in case the answer is Yes, one in case the answer is No and each arrow leads us to the second question

Note: In this exercise, you want to start with a "non optimal" question that will create a longer tree.

Tell the students: "Let's start with the question: Is the shape red?"

Draw the following decision tree on the board



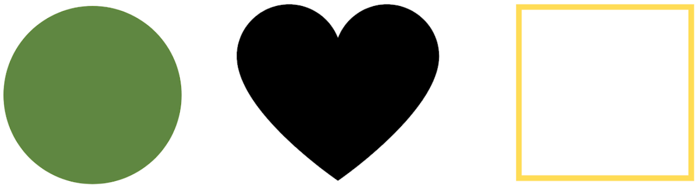
We represent a question using a diamond shape

Ask the students what happens in case the answer is yes?

-> Because only 1 of the 4 possible shapes is red, we have found our answer. We draw that in our decision tree using a “parallelogram”

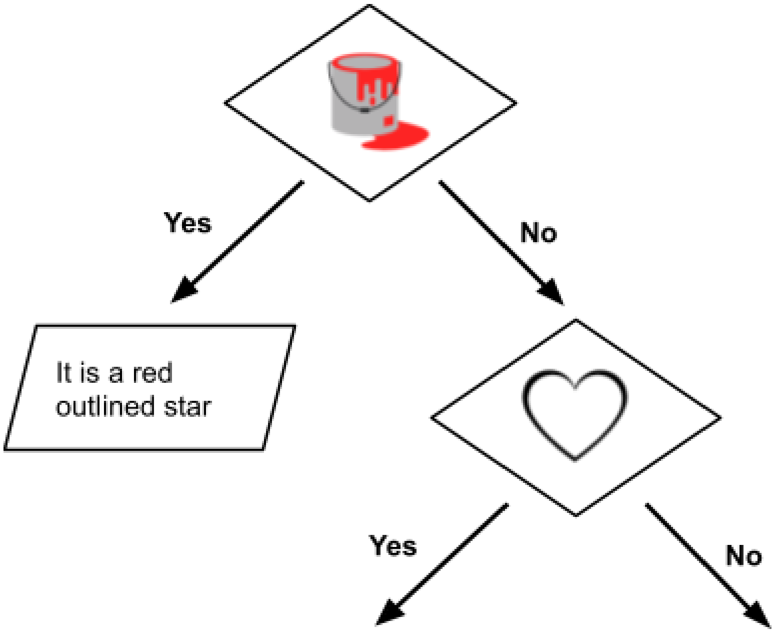
Ask the students, what can they deduct in case the answer is no?

-> Because only 1 of the 4 possible shapes is red, we can eliminate it and we are left with the following shapes

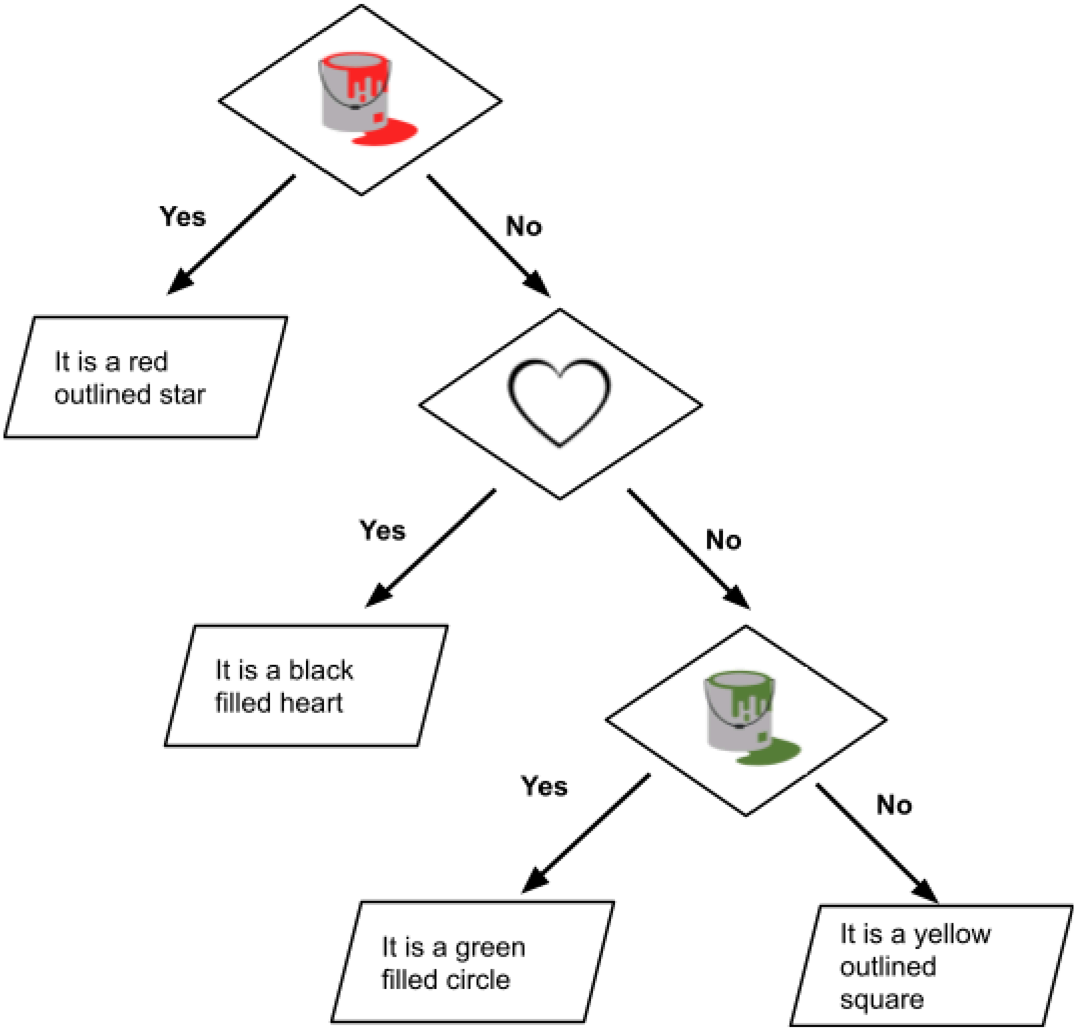


What could be our second question? Take a suggestion from the students. Let’s suppose our next question is “Is the shape a heart”?

Our decision tree becomes



Ask students what happens in case the answer to the second question is Yes, what if it is No? Draw the rest of the decision tree with them. A possible final decision tree could be:


Yes
No
It is a red outlined star
Yes
No
It is a black filled heart
Yes
No
It is a green filled circle
It is a yellow outlined square

Reflections:

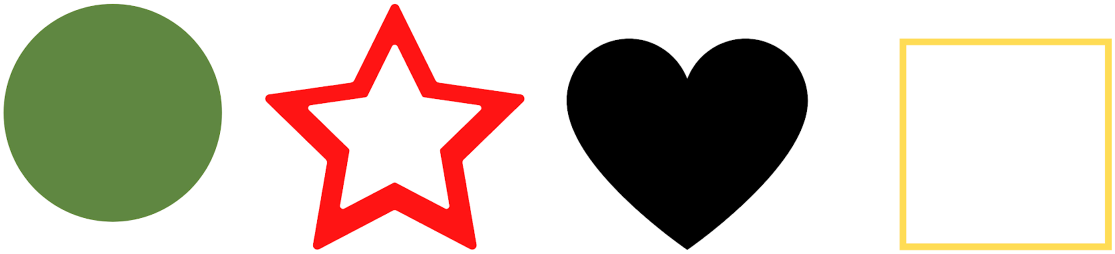
- If we are very lucky, how many questions does it take us to get to the correct answer?->1
- If we are not lucky at all, how many questions does it take us to get to the correct answer? -> 3

We call our drawing a **decision tree** because it looks like an inverted tree growing as we ask more questions. If we want a computer to learn how to guess the correct shape, we would write the instructions following the tree. Because the tree has a specific order, it represents our **algorithm**. An **algorithm** is a set of instructions given in a specific order so that a computer can solve a problem

Well, bad news: the computer tends to be very unlucky. And each time it needs to ask a question from the decision tree, it requires you to feed it a candy. You do not want to give away too much candy, so let’s see if we could make a better decision tree.

3. Making a better algorithm

- Put the students in groups
- Distribute Activity 1: Finding the first question
- Ask them to
    - Find the differences between the shapes
    - Find the common patterns



Regroup the students and ask them to share their findings. Did they find a pattern that separates the shapes? -> Filled/Outline. What is the superpower you used to be able to get to this answer? -> **Pattern Recognition**

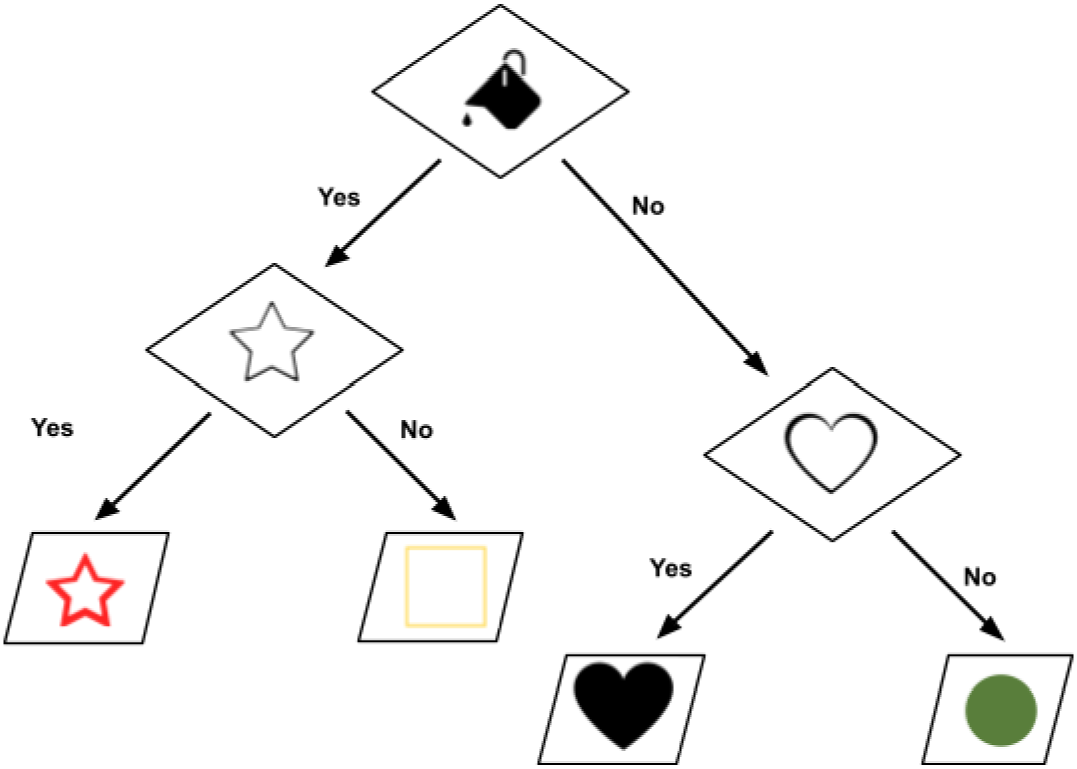
You need to look a the set of yes/no pattern questions that would be helpful to separate the shapes into 2 groups: filled or not filled (outlined). I will give you 3 questions to choose from:

- Is the shape green?
- Is the shape a square?
- Is the shape filled?

Which ones should we eliminate and ignore? Which one should be kept? (-> Is the shape filled?) When you decide what pattern matters the most and which one to ignore, you are using another superpower called **“Abstraction”.** Abstraction is to keep the details that matter the most and ignoring what is not important.

Draw with them the decision tree with the question starting with “Is the shape filled?”

The possible result could be:

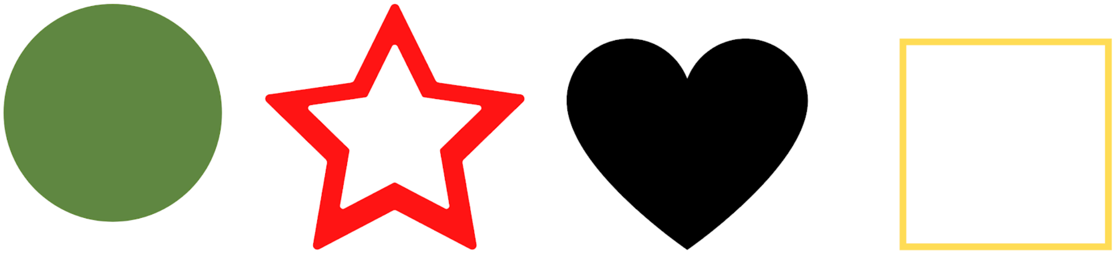


4. Wrap Up and Reflections

Reflection Points:

- Can you notice something different about our new tree?
- What is the maximum number of questions our computer could be asking? Is that better or worse than in our first tree?
- What was one advantage of finding a pattern?
- What if we had found several patterns: for example filled/outlined and the green color, how do we decide which one to ask first? -> the one that eliminates the most shapes
- Creating a decision tree is similar to creating a list of Yes/No question to solve the problem "Guess my card". This is also called an **Algorithm**
- Creating a "better" tree can also be referred to as making an "**efficient**" Algorithm.

Student Activity 1: Finding the 1st question



1. What is different about the shapes?

____

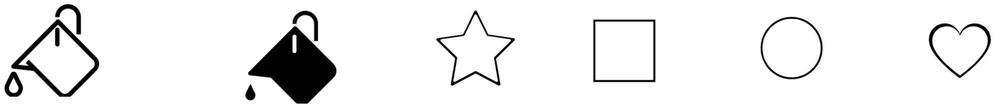
____

____

____

2. Can you find common things between at least 2 shapes?

Circle the pattern(s) you find:



Outline Filled Star Square Circle Heart