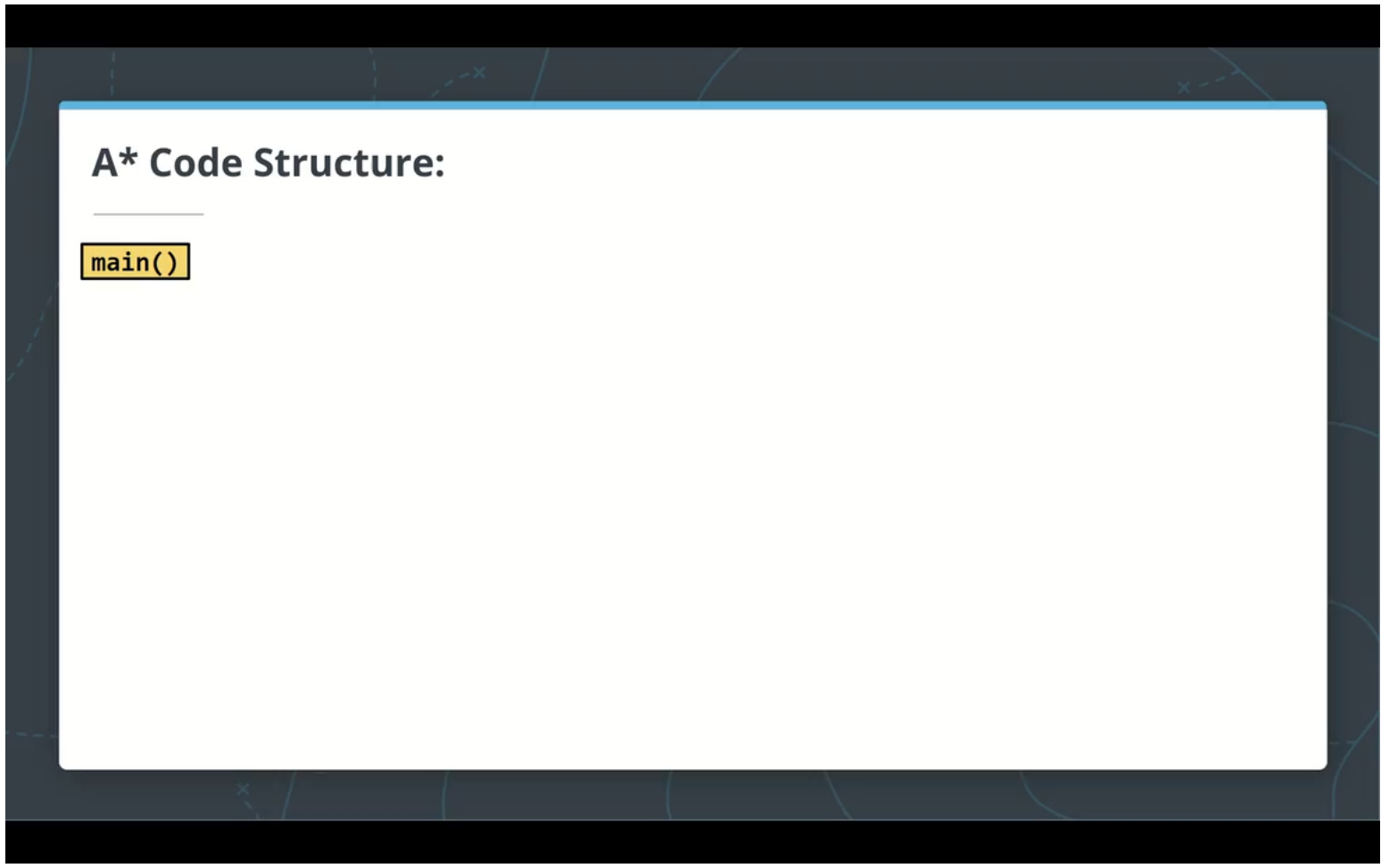
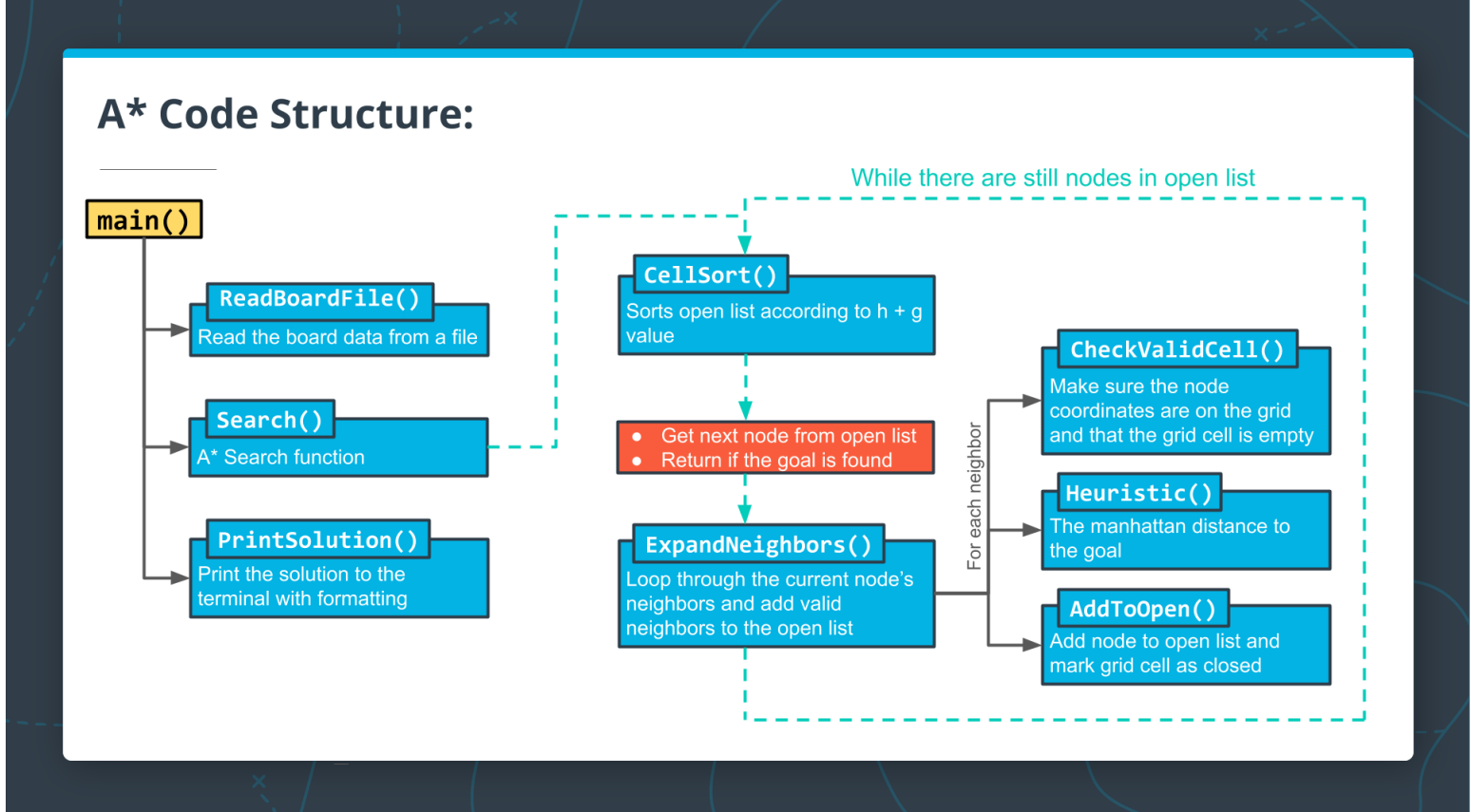


### Lesson Code Structure

The next video provides an overview of the code that you will be implementing throughout the rest of the lesson. Be sure you have carefully reviewed the pseudocode from the previous concept so you'll have a better understanding of each of the functions described in the video.



[https://video.udacity-data.com/topher/2019/August/5d4a1831\\_ascii-a-star-first-take/ascii-a-star-first-take\\_720p.mp4](https://video.udacity-data.com/topher/2019/August/5d4a1831_ascii-a-star-first-take/ascii-a-star-first-take_720p.mp4)



#### Quiz

Below are the steps from the `while` loop in the A\* pseudocode you saw previously:

**while** the list of open nodes is nonempty:

- Sort the open list by f-value
- Pop the optimal cell (called the *current* cell).
- Mark the cell's coordinates in the grid as part of the path.
- If the *current* cell is the goal cell:
  - return the *grid*.
- else**, expand the search to the *current* node's neighbors. This includes the following steps:
  - Check each neighbor cell in the *grid* to ensure that the cell is empty: it hasn't been closed and is not an obstacle.
  - If the cell is empty, compute the cost (g value) and the heuristic, and add to the list of open nodes.
  - Mark the cell as closed.

In the quiz below, match the steps to the appropriate function from the A\* code structure diagram.

QUIZ QUESTION

Match the steps in the `while` loop of the pseudocode above to the functions in the code structure diagram where the code would be implemented.

Submit to check your answer choices!

FUNCTIONS	STEPS
<code>CellSort()</code>	1
<code>ExpandNeighbors()</code>	5
<code>Search()</code>	4

SUBMIT

#### Summary

The code for the A\* search algorithm has been broken down into the following functions:

- `CellSort()` - sorts the open list according to the sum of  $h + g$
- `ExpandNeighbors()` - loops through the current node's neighbors and calls appropriate functions to add neighbors to the open list
- `IsValidCell()` - ensures that the potential neighbor coordinates are on the grid and that the cell is open
- `Heuristic()` - computes the distance to the goal
- `AddToOpen()` - adds the node to the open list and marks the grid cell as closed

You will be implementing these functions along with a few other small helper functions throughout the rest of this lesson to complete the ASCII A\* search program.