Stakeholders

Despite every effort is made to ensure this game is as interesting as possible, this game is not designed for absolute beginners who want to learn more about what computer science is, the reason of which is that this game assumes basic knowledge in computing such as a tree in graph theory and it does not include introductions to these theories. If a user wants to learn more about the implementation of an algorithm, programming knowledge may be needed. However, it is still possible for a learner to get a taste of advanced algorithms, although this game is not designed to do so. In conclusion, there are two main possible types of stakeholders:

1. Those who have already obtained some basic concepts of algorithms and want to study further in this area, then this game would be a good starting point to get a feeling;
2. Teachers might want to utilise this game in lessons while introducing relevant topics in order to add more fun to lessons. As each level in my game will centre around a key algorithm, e.g. dynamic programming, then this game can be played when DP is going to be the topic of the lesson.

Research

Coding game

Creating interesting game background to help leaners learn coding from scratch

Competitions similar to OJ but multi-player

Different types of games to improve coding skills

Lacks knowledge of some advanced algorithms

Fancy visual effects are used as an illustration of some rather abstract ideas in programming which makes this platform quite fascinating for new coders who want to make a start. However, most of the content is for beginners who rather than experienced programmers. They also offer multi-player competitions where one can challenge friends, schoolmates or coworkers, which is fun and engaging, but there is less visualisation in some of the higher level problems. In addition, those coding challenges are less difficult than those on lots of OJ platforms.

<https://www.codingame.com/start>



[CS-Playground-React](http://cs-playground-react.surge.sh/)

A simple in-browser JavaScript sandbox for learning and practicing algorithms and data structures, lots of classic sorting algorithms as well as data structures that are frequently used are included in this platform. Some challenges are provided at the end to test one’s knowledge and make it more fun. In addition, it also offers solutions when one gets stuck, and comes chock-full of links to helpful articles, tutorials, and other resources but that is provided as an external link which makes it less interactive. In conclusion, this website does teach you some very important data structures and algorithms but not in a game context.

<http://cs-playground-react.surge.sh/>



Teaching Kids Computer Programming

There are lots of introductory games for kids as a taste of programming, for example, the Scratch programming language. And this website also offers 6 free games for teaching kinds programming. Although these games are very interesting and easy to play, they are designed to be a very brief introduction to programming which is not what I intended to do.

<https://educators.brainpop.com/2014/09/26/6-free-games-teaching-computer-programming-kids/>



Online Judging system

<https://uva.onlinejudge.org/>

<https://leetcode.com/>

<http://codeforces.com/> 

These are all very well-known online judge systems where programmers can practice solving algorithmic problems and submit their solutions to judging systems. Feedback can normally be given in a few seconds. Some competitions may also be held regularly on these platforms. They are all perfect places to improve coding skills, but those problems are all designed by experts and tend to be very challenging. And it is often the case that there is no visualisation of problems available on these websites, which makes solving these problems even harder.

UK Bebras challenge

This challenge is held annually and introduces computational thinking to students in different age groups. The main part of this challenge is to solve puzzles that require logical thinking rather than prior knowledge in computer science. Web-based human interaction is also available so that the participant can interact with the computer to obtain the solution.

<http://www.bebras.uk/students.html>



Ticket to Ride – a board game

*Ticket to Ride* is a cross-country train adventure where players collect cards of various types of train cars that enable them to claim railway routes connecting cities in various countries around the world.

Students through the missions that they choose about connecting one city to another, come across the implementation of some path finding and minimum spanning tree algorithms, such as Kruskal’s, Prim’s and Dijkstra’s algorithm.

<https://www.daysofwonder.com/tickettoride/en/>



An Educational Game for Teaching Search Algorithms

This is an article about using a Pacman game designed by the university team to teach searching algorithms such as DFS, A\*, etc. This game consists of detailed explanations and visualisations of these algorithms in a game context. Students who play this game will also have the chance to apply their knowledge in this game in order to solve some of the challenges in this game.



Brief conclusion:

Having done the research, I understand the fact that there are some similar platforms and projects are committing to the visualisation of algorithmic problems, though with various approaches and levels. For instance, coding game has fascinating game context for new coders and makes it much more appealing to beginners, while online judging systems provides much more challenging and well-designed problems and feedback on users’ solutions after they submit their codes. If we go back to my initial thought, I intended to combine interesting stories and abstract computational ideas together to make the learning process less painful. This idea is similar to the game mentioned in the article An Educational Game for Teaching Search Algorithms. The game comes with explanations of great details and vivid illustration. Human interaction is achieved by asking students to apply what they learned in another mode. This is the closest solution to my problem and thought, but there is still something to add.

In this project, only searching algorithms are included and they are all demonstrated in the same game. (which does have some advantages) I want to add more algorithms of different categories and different games might be used to introduce different algorithms. It will consist of three features:

1. The game is a combination of OJ and Pacman for teaching search algorithms, two of the existing solutions to my problem. Therefore, most of my games will be based on problems from online judging systems. In other words, OJ problems will be visualised in my game. This is due to the fact that lots of problems from OJ are described as games themselves, which already adds fun to my game and makes it easier to adapt them. In addition, qualities of problems from OJ are generally very good and I will also try my best to select those classic ones to ensure leaners can have a good learning experience.
2. There are a range of games included in my project and most of them will have different levels of difficulty. Brute force may be able to address some of the easiest levels but will become too inefficient to be used when coming across harder levels. This design is based on the idea that efficiency is usually the key part of an algorithm design when the data scale becomes very large, which is often the case in the real world.
3. Last but not least, this game will be able to run offline. Bizarrely, nearly all existing solutions require users to have a stable Internet connection. Although publishing a game online will make it accessible to more people, being able to run offline gives users the opportunity to access it anywhere and anytime they like.

Limitations:

Visualizations can demonstrate the operational functionality of algorithms and are designed in line with the principles of student’s active learning