Accredited

A LEVEL

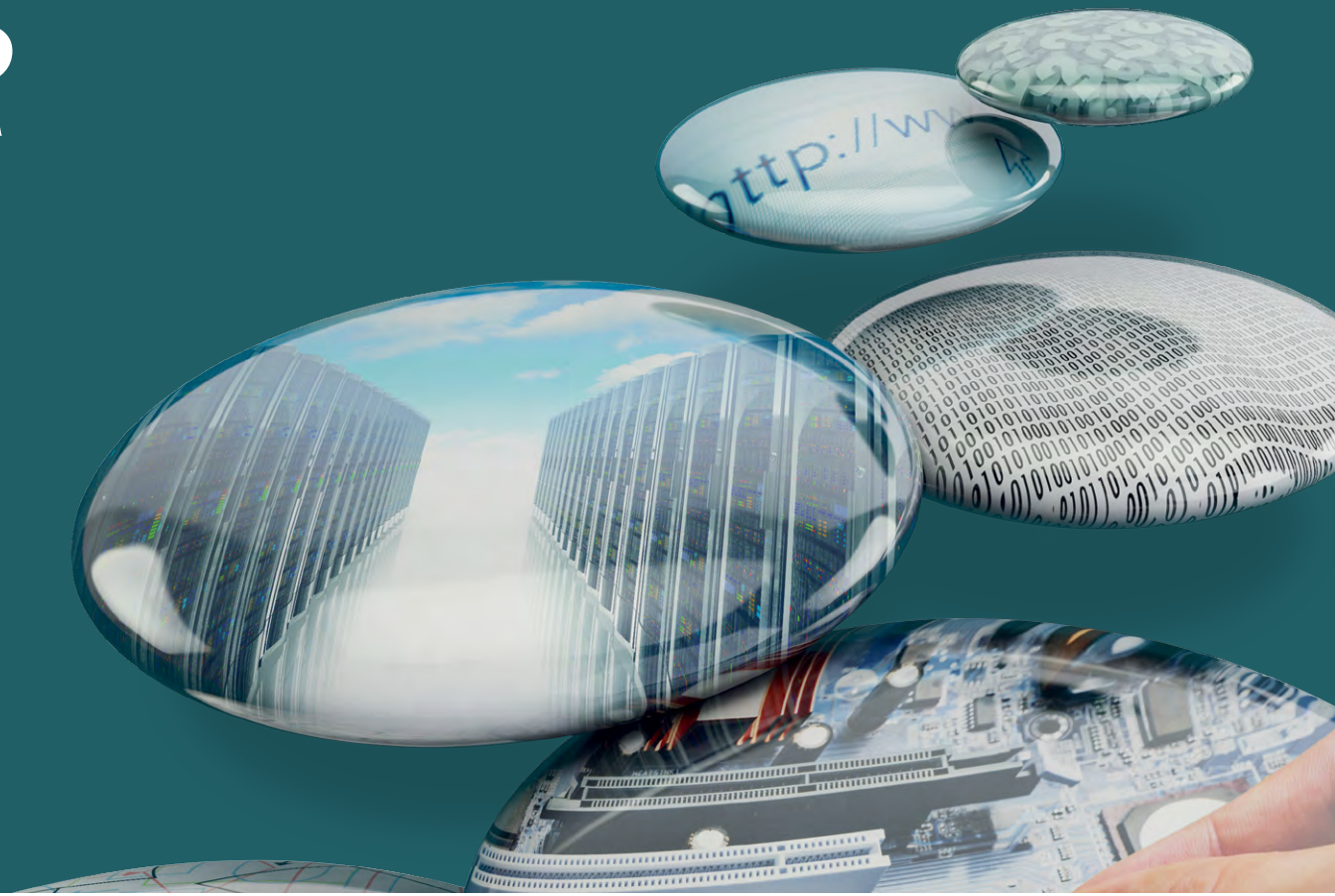
Project Complexity Guide

H446

COMPUTER SCIENCE

Theme: Project complexity guide

September 2015



CAMBRIDGE UNIVERSITY PRESS
www.cambridge.org

OCR
Oxford Cambridge and RSA

We will inform centres about any changes to the specification. We will also publish changes on our website. The latest version of our specification will always be the one on our website (www.ocr.org.uk) and this may differ from printed versions.

Oxford Cambridge and RSA Examinations is a Company Limited by Guarantee.
Registered in England. Registered company number 3484466.

Registered office: 1 Hills Road
Cambridge
CB1 2EU

OCR is an exempt charity.

Contents

Project Complexity Guide

Introduction

The A-Level project gives candidates the opportunity to go through the process of developing a substantial piece of software. In order to be able to access the full range of marks available candidates need to have a sufficiently complex project. This guide aims to give an overview of what makes such a project.

Advice on complexity in projects with examples

Depending on the requirements of the project some of the features below will be more relevant than others. These aren't meant as a tick list but as guidance to some of the features a sufficiently complex project will have. For example, a project may involve complex algorithms and make good use of external libraries but only have a few settings that need storing in a simple text file. The important test of whether a project has the required complexity will be if there is enough depth for a candidate to make meaningful decisions about its development and to build it through meaningful iterations.

Storing Data: Most programs will need some form of persistent storage. This may be as simple as storing user preferences or may be as complex as storing large amounts of data to be analysed and generate reports. Candidates will be expected to choose suitable methods of storing this data. In some cases a simple text file or CSV is sufficient in others it may be appropriate to use such storage methodologies such as JSON, XML or even a relational database. These latter options can be indicative of a more complex project.

Use of Libraries: Libraries allow candidates to add functionality to their projects that they may not have the time or expertise to code themselves. Candidates should not worry that they will be penalised for the use of libraries, on the contrary, use of libraries to tackle parts of their problem is best practice. Sensible choice and use of libraries can be an indicator of a good level of complexity in a project. *Candidates should be reminded to clearly reference where libraries are used.*

Programming Style: There is no set paradigm for the project. The majority of candidates are likely to choose a procedural approach but others are equally acceptable. Some projects may be suited to object-oriented, functional, logical or even assembly programming. Whilst by no means a requirement a candidate choosing a particular paradigm is a good indication that the project has a good level of complexity. Where the project has scope for the candidate to demonstrate commonly used techniques in their chosen methodology this too is an indication of a good level of complexity. For example, a candidate programming in an object-oriented language may structure their program using a model-view-controller pattern, a candidate creating a web-based project may use AJAX for improved responsiveness.

Algorithms and Data structures: There isn't an expectation for candidates to reinvent the wheel. If a program requires sorting of a data structure then a candidate deserves no less credit for using a language's built in sort routine (as a professional programmer likely would) than if they code their own quick sort algorithm. Similarly if a required data structure is offered by a language a candidate would be well advised to use them rather than recreating them. This said there will be the need for algorithms that aren't available natively. These might be established algorithms such as Dijkstra's Shortest Path or MinMax which the candidate needs to implement or an algorithm the candidate has had to devise for themselves. Likewise, existing data structures may need to be adapted to best suit the project. A candidate needing to put clear thought into developing their data structures and algorithms and using them alongside or integrating them with pre-existing data structures and algorithms is indicative of sufficient complexity.

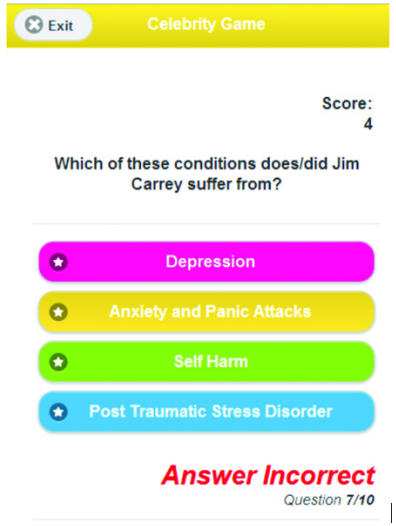
Other: There are many other indicators that can show complexity. These include:
– The need to write code to interact with a specific hardware device.
– Writing code so devices can share data across a LAN/the Internet.
– Combining different technologies in order to produce the final product.
Candidates may need guidance at the very early stage of their project as to whether it has enough scope to be complex and if not how its complexity can be added.

An Example

A student decides to make a mental arithmetic quiz for young children. It will store the scores each child gets when taking a test.

This as it is, is unlikely to be complex enough to make an A-Level project. Indeed this is what might be expected of a GCSE task.

That's not to say the project isn't viable, it just needs a bit more depth. If the student were able to allow the system to adapt the questions it asks according to previous performance and graphically show the teacher progress each child's over time here would then much more scope to demonstrate the appropriate skills.



This could include:

– Storing the children's performance in a relational database.

– Devising an algorithm to continually 'learn' which questions the children need more help with.

– Using a library to generate graphs for the teacher.

Description of five (good) example projects (real) and description of how they are complex enough for A Level

The following projects are examples that were submitted for the previous specification, but demonstrate the necessary complexity for the current specification.

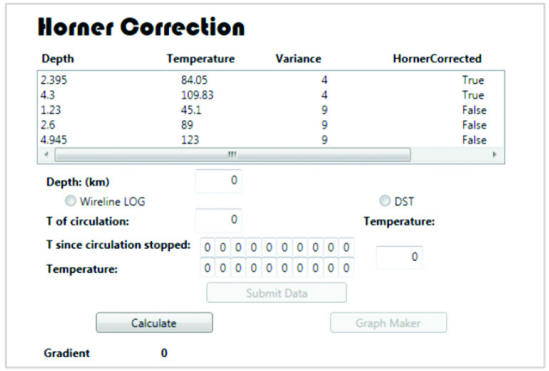
Mental Health Awareness App: This student was asked by a local council to develop an app to increase awareness of mental health issues. The student built an app that allowed the user to access information regarding mental illness and allowed them to take quizzes to test their awareness of mental health issues. He wrote a content management system so the council could update the information and quizzes when they wanted.

(Indications of complexity: On first inspection the project may not seem complex but further investigation shows a lot was going on in the background. The addition of the content management system already takes this beyond a simple quiz. Whilst the app was effectively a web page it needed to be useable when the user's device was 'offline'. The student researched how to create a local database to store the data necessary for the app then got this to sync with the database council used for their content management system when the app went online.)

Stealth Game: This student had a particular enthusiasm for stealth-based games. Using some peers as a user group he created his own game which involved sneaking past guards, in increasingly complex levels, to recover different objects. Each of the guards moved around the level and had their own line of sight affected by things such as lighting and objects being in the way.

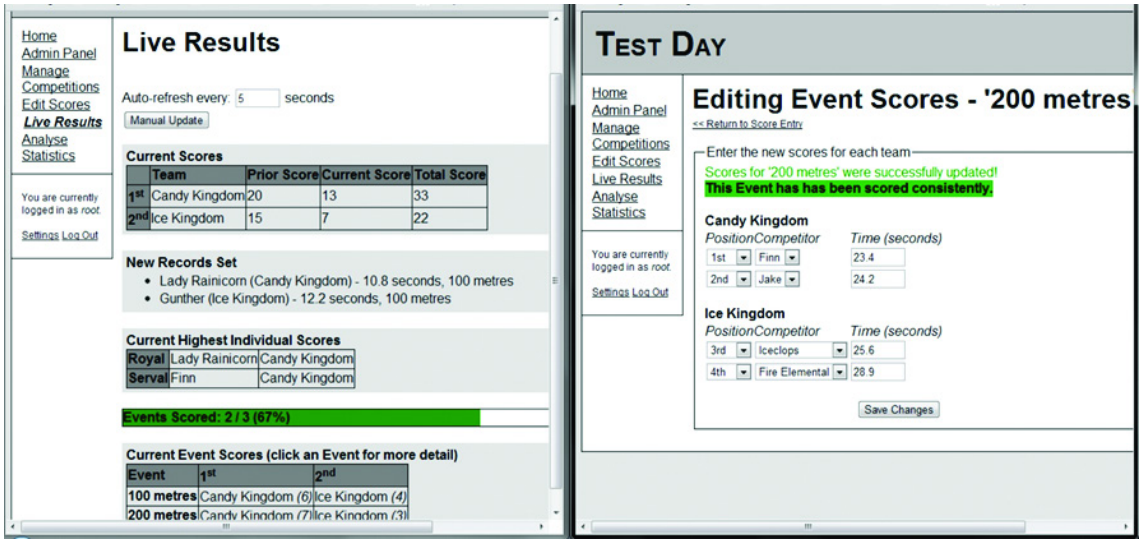
(Indications of Complexity: The student investigated and adapted existing algorithms to simulate the guards' field of vision. This involved a significant amount of research and prototyping. The game was coded in C# using the FlatRedBall game engine. The settings and high scores were stored in an XML file.)

Horner Correction: This student had previously undertaken work experience with a company that carries out 'Geochemical Investigations'. One of the tasks the company does is analysing temperature data from boreholes to determine if it is at the bottom of them. This is done through a complex set of calculations called Horner Correction. The student wrote an application to perform this analysis. What particularly stood out in this project was that she adhered to the company's software engineering guidelines whilst doing so. The project was developed using the model-view-controller pattern and was set up in such a way that it could be analysed using the company's automated testing software.



(Indications of complexity: The front end of the system is relatively simple but there was a great deal of complexity underneath. The project itself was coded using an object-oriented approach following an MVC pattern. The calculations required use of matrices and statistical calculations. The student integrated the functionality of specialist mathematical libraries into her own coding of the Horner correction algorithm and used a graph library to help produce easily readable reports.)

Sports Day Scoring System: The student's school takes Sports Day very seriously. Records are kept of each year group and event and trophies awarded to the most successful of the school's four 'Houses'. In previous years they had used a set of spreadsheets to keep track of the scores but this was error-prone and required a lot of effort to be duplicated. This student developed a web-based score keeping system. The system was coded in PHP and ran off a computer acting as a webserver on the LAN. This meant that multiple students could update the scoring system as results for events were coming in. The system was able to instantly show whether any school records had been broken and how each of the school's 'Houses' were performing. The school's announcer was able to log onto the system on his tablet and access automatically updated results.



(Indications of complexity: The results were stored in a relational database. The front end used jQuery library in conjunction with for JavaScript code. AJAX was used to help improve the responsiveness of the application.)

Ethical Coffee Game: The student user was a member of an ethical group promoting the purchase of coffee that ensures profits are shared with the growers. The student created a game which involved moving ingredients around a grid and combining them in sequence in order to build different coffee-based drinks. Players had a limited number of moves and were given extra credit for using ethical ingredients.

(Indications of complexity: The student took an object-oriented approach to the project. They used external libraries to help deal with the sound. They created their own data structure based on nested arrays, HashMaps and detailed algorithms for the game's mechanics.)

What is not complex enough for A Level?

Using applications such as Access or Excel and adding scripting will not offer the complexity necessary for an A-Level project as they can be overly restrictive and do not give the candidate the opportunity to demonstrate the required decision-making skills. Often too much of the functionality that would be expected to be developed by the candidate is built into these applications. Candidates are expected to produce a stand-alone solution; this is not possible when using these applications.

Web-based projects have the potential to have more than enough complexity for A Level but caution needs to be taken that they don't fall into the trivial category. Needless to say a static HTML site would warrant no credit. It is, however, quite acceptable for a student to code their program in JavaScript if it's, for example, a game. Just because a site is dynamic that does not in itself make it complex. A site which just has a login page and perhaps pulls some content from a database is not in itself likely to be complex enough.

Visual drag and drop languages such as Scratch, AppInventor and Flash BYOB are not suitable to produce a program for an A-Level project (though there is nothing stopping a student using them for prototyping algorithms in a project's early stages).

List of potential areas to explore with projects:

Students should be encouraged to find real users in need of an application and where possible to combine their project with their own interests. Potential project areas might include:

Games

Management Information Systems (e.g. room booking systems)

Simulations of physical systems

Apps (e.g. fitness/diet trackers)

Content Management Systems

We'd like to know your view on the resources we produce. By clicking on the 'Like' or 'Dislike' button you can help us to ensure that our resources work for you. When the email template pops up please add additional comments if you wish and then just click 'Send'. Thank you.

If you do not currently offer this OCR qualification but would like to do so, please complete the Expression of Interest Form which can be found here: www.ocr.org.uk/expression-of-interest

Version 1

OCR customer contact centre
General qualifications
Telephone 01223 553998
Facsimile 01223 552627
Email general.qualifications@ocr.org.uk