

CS25210 Coursework 2014:**An HTML5 Canvas game using animated sprites.**

This work will be assessed according to the assessment criteria AA ("Development").

<http://www.aber.ac.uk/~dcswww/Dept/Teaching/Handbook/AppendixAA.pdf>

Part 1 (web based component) is due in on Friday 4 April on or before 11:59pm. Part 2 (report) is due in on Friday 11 April on or before 11:59pm . The assessment for this module is 100% coursework which is broken down into 80% implementation, 20% report. Both parts should be submitted online via Blackboard. This is an individual piece of work and the usual departmental regulations concerning plagiarism apply.

Web-based component (80 marks)

This should be an online game, written using HTML5 and JavaScript. The marks breakdown for this will be as follows:

Website (5 marks): The game should be incorporated into a small website created in HTML5 and styled with CSS3. This must include an "about" page and a "play game" page. The "about" page should include credits if you have used any public domain libraries, images, sounds, or similar. If you wish to have other pages feel free to add more.

Core game (20 marks): This will be an interactive game involving graphics presented using the HTML5 canvas element; it will require mouse interaction to control the gameplay. The variety of game is up to you – space invaders, pacman, asteroids, dodge the blocks... there are any number of possibilities. A very basic game will pass, but not much more: what we want to see is creativity – think animated graphics, levels, etc.

Playability (15 marks): The game should have defined start and end states (e.g. "click to start", "You scored X, click to play again"). The motion in-game should be smooth. There could be levels of some kind – getting faster or more difficult as the game progresses.

Graphics (15 marks): Graphics should be of the appropriate type. At least one game character ("sprite") should be animated such as the bitmap car featured around slide 20 in Lecture 2. You do not have to create the sprites yourself, but if you use sprites found online you must include the location of the sprites and a credit to the originator/designer/artist on your About page.

Alternate controls (5 marks): Incorporate keyboard controls as well as mouse controls.

In-game help (5 marks): The game presents tips as you are playing, or an in-game help screen activated by pressing a key/clicking on a button.

Sounds (5 marks): Sound should play during in-game events. If you choose to have music playing constantly in your game, please make sure there is a control to turn it off.

High Score (10 marks): A score should be kept. For full marks this score should be persistent between visits, and should be achieved using HTML5 localStorage.

Submission formats

The web-based element should be submitted in two ways:

1. All files (website, any files required for the game) should be enclosed in a zip archive. It is your responsibility to check that all client-side parts of the game and site work as extracted.
2. In addition you must submit a URL by email to hmd1@aber.ac.uk which points to a working version of your submission, hosted on your Aber homepage.

Note: if you have any server side code in your implementation of this coursework it is the online version that will be used to test this - you do not need to get this part working from a zip archive, but you must submit the code. There is **no requirement** to do any server side code for this module, and no marks available either. Some students like to exercise their web skills, however, and that is fine - there is no penalty for using PHP or any other server side technology.

Written component (20 marks) :

This concerns the game, and will have four sections given equal weight:

Executive summary giving an outline of the game structure, including storyboards where appropriate. This section should describe the design choices and game mechanics without reference to implementation details, and should be easy to understand for someone who's never programmed before.

Technical overview providing justification of the choice of technology for this game. This section should explain, for example, why produce the game client side; what the technological alternatives would have been (e.g. Silverlight/JavaFX/Flash); what libraries you have used if any (e.g. "jQuery was used for..."); if you have chosen to incorporate server-side elements, discuss these too.

Software testing featuring cross-browser and cross-platform testing results. Your game does not have to work perfectly on every browser and every platform, but you should be aware of where it fails and be able to discuss the issues. This section should include screenshots where appropriate. For example – does your game work on your telephone (or the telephones of your friends)?

Reflections and future work which should detail how the game could be improved and extended. Are there extra levels you could build? Is there something you wanted to get working but ran out of time? Could a different technology give you more freedom? Could the game make money?

The report should be submitted in **PDF** form only. As a guideline, it should contain around 1000 words, with no limit on the number of screenshots or illustrations. This is **short**, leaving no room for waffle. A good report will be clear, concise and well-illustrated.

Note: Overly long reports (>1200 words) will be penalised, as will reports submitted in formats other than PDF.