

School of Computer of Science

ASSIGNMENT BRIEFING SHEET (2018/19 Academic Year) – ANONYMOUS MARKING

Assignment Title	Mobile App Development Using Xcode	Submission Date	07/01/2019, Monday
Module Title	Mobile Computing	Module Code	6COM1047
Tutor	Xianhui Cherry Che	GROUP or INDIVIDUAL Assignment	Individual

FOR INDIVIDUAL ASSIGNMENTS – STUDENT TO COMPLETE

By completing **BOX A** below, I certify that the submitted work is entirely mine and that any material derived or quoted from the published or unpublished work of other persons has been duly acknowledged. [ref. UPR AS12, section 7 and UPR AS14 (Appendix III)]. I also certify, that any work with human participants has been carried out under an approved ethics protocol in accordance with UPR RE01.

Please **ONLY** provide your ID (sm) number as this assignment will be anonymously marked

BOX A

Student ID Number (SRN)

15016466

Permission

The School of Computer Science may include your work in a showreel, which will be presented in various public occasions to demonstrate the students' skills and abilities from University of Hertfordshire. By signing your name below, you give consent to the School to include your work in the showreel.

Student Signature: muhammad sarim (Name print is fine.)

**This sheet must be submitted with the assignment, and either BOX A filled in.
LATE SUBMISSION WILL ATTRACT A STANDARD LATENESS PENALTY.**

1. For undergraduate modules, a score of 40% or above represents a pass mark.
2. For postgraduate modules, a score of 50% or above represents a pass mark.
3. For work submitted up to 5 working days late marked is capped to a bare pass (40% for undergraduate and 50% for postgraduate).
4. For work submitted more than 5 working days a mark of zero will be awarded for the assignment.

ASSIGNMENT BRIEFING SHEET (2018/19 Academic Year)

ANONYMOUS MARKING

THE ASSIGNMENT TASK:

The task of this assignment is to develop an iOS mobile app using Xcode platform. The app will be a type of third-person adventure game, as indicated by the Fig. 1 and 2.

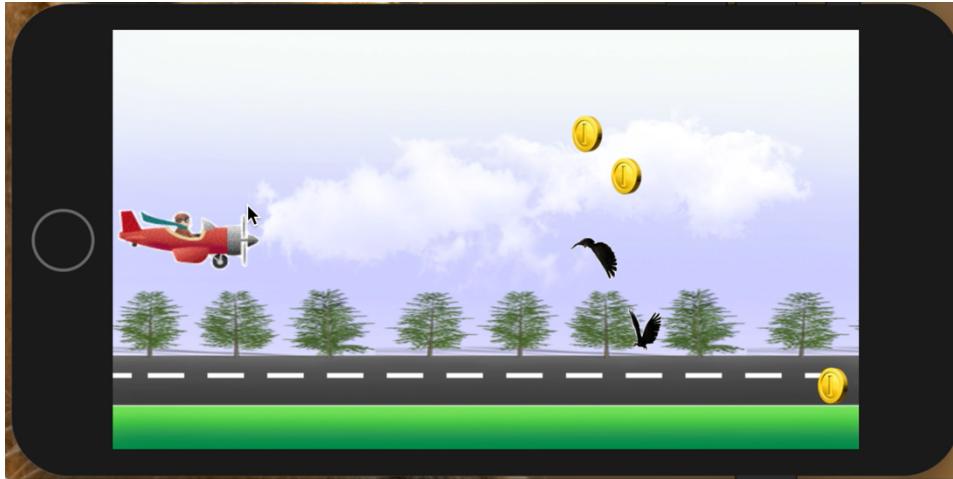


Figure 1. Gaming Scenario

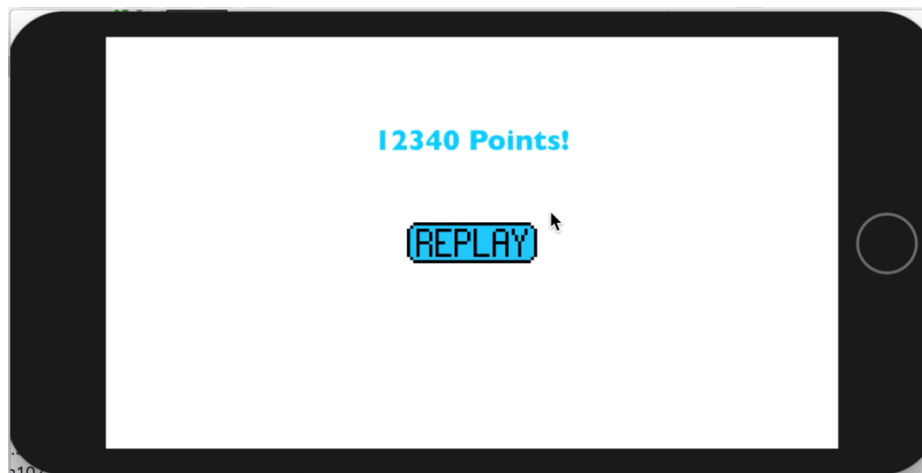


Figure 2. Completion Scenario

Core Functions:

- The app should be used in the landscape mode only.
- None first-person perspective.
- The main avatar is the airplane as shown in Fig. 1. The motion effect should give the users an impression that the airplane is travelling along the road, e.g. from the left to the right.
- User can use finger to drag and move the airplane within the screen, indicating acceleration or deceleration, rising or declining motion. The movement should be kept within the screen boundary.
- There are two types of objects that will randomly appear from the right side of the screen.
 - The crows are obstacles that the airplane should try to avoid.
 - The coins are bonus that the airplane should aim to hit.
- Upon the collision with the two types of objects, there should be appropriate motion effects, and scores should be updated accordingly.

- Each game play should last no more than 20 seconds. When the game is over, display a completion screen, with the score information and replay button facility.

Usability Requirements:

- Target users are casual game players.
- The app should be easy enough for the target users to play on their own, and appealing enough to attract repeated play.
- The app should be able to run on any iPhone models of any screen size. (The latest models iPhone XS and iPhone XR may not be available on the iOS simulator in the lab, therefore they can be excluded in the assessment.)
- It is not mandatory to use the theme given in Fig. 1 and 2. You may choose any theme that you prefer in order to increase the appeal of the app.

Demo Video:

You can view the showcase video of this app on [Canvas/Units/Assignment/Showcase Demo.mov](#), which gives you an idea of the app design. In reality, you do not have to implement the app that is as identical as the one shown in the demo video – if you have better themes, you are welcome to give it a try. Regardless the theme, the core functions must be met.

Upon completing the coursework, you need to produce a demonstration video of your app and submit it along with your source code. [Please read Canvas/Units/Assignment/Video Recording for guidance of recording videos](#). Please note the quality of your demo video will not be assessed. The demo video will serve as a crucial evidence of your work. Should your app fail to run during the testing in a different environment, your work will not be capped at 40%. The quality of the demo video will not be assessed.

Production:

Should you wish to, you can add some bonus features such as sounds or levels. But do NOT be too ambitious. You should rather have a simple app that works, than a sophisticated app that does not work. Please bear in mind this is a production process – if you do not have a product in the end, you will fail the project.

As this is an assessment of production, failure to deliver a working product will fail the assessment:

- An app that fails to run due to poor file management (such as missing files) will receive a penalty of 5 marks. Excuses such as last-minute rush in submission or problems with file-zipping will not be accepted.
- An app that fails to run for various other reasons will be capped at 40% for marks.

Your production should be managed by Agile and version control, as explained in [Lecture 2](#). Do NOT wait until the end of the project to conduct management activities. Failure of producing consistent evidence of managing the production throughout the project will receive penalties.

Resources & Copyright:

The images used in the showcase video are available on [Canvas/Units/Assignment/images.zip](#). Please feel free to use them.

Please bear in mind that this is not an art module, so you are not expected to spend much energy on graphic design. Using images and audio files from other online sources is acceptable, but you should acknowledge the source in the completion page. For example, using a small print in the corner of the screen to reference the source.

Also note that your apps are strictly used for the purpose of internal study only, as some apps may incur potential copyright issues with image or sound sources. Under no circumstance should you be permitted to publish your apps in the public domain under the name of the University. (You are welcome to demonstrate your app during a job interview.) Please consult the module team if you have any doubts.

Development Strategy:

There may be many approaches to implement this game. Here is one possible pathway for the essential development:

1. Create a roadmap background with motion effects.
 - a. [Read Lab 5 for using CA to create animation with alternating images.](#)
 - b. [Read Lab 3.2 for screen-fit programming to make sure all images are sized and positioned properly.](#)
2. Place an airplane in the sky, and make the main avatar image object drag-and-move.
 - a. [Read Lab 4 for making drag-and-move images.](#)
 - b. [Read Optional Tutorial – Constrained Movements with Imaging Dragging for how to move the object within boundary.](#)
3. Create a series of obstacle crows that will randomly appear from the right of the screen. Forget about coins at this stage. You can implement coins after successfully implementing crows.
 - a. [Read Lab 3.1 for creating a new image view programmatically.](#)
 - b. [Read Lab 6 for generating delay.](#)
4. Make the obstacle crows move to the left of the screen, preferably with various speed.
 - a. [Read Lab 6 for using UIKit Dynamics to create falling motion with linear speed.](#)
5. Give the crows motion effect so it looks like flying.
 - a. [Read Lab 5.1 for using Core Animations to create motion effect with changing images.](#)
6. Create collision between the airplane and crows. For the time being, just leave the airplane static during the testing process.
 - a. [Read Lab 5.2 for how to add collision behaviours. All obstacles need to be added to the items of collision behavior. As for the main avatar, it is a bit tricky. You probably have thought of two options:](#)
 - i. [Like obstacles, the main avatar is also added to the items of collision behaviour;](#)
 - ii. [Add the frame of the main avatar to the collision boundary of the obstacle cars.](#)
 - b. [It is advised to use option \(b\) instead of \(a\), as it is more stable. Since the main avatar is associated with drag-and-move action, it will not react to the collision behaviour dynamics effectively.](#)
 - c. [Read Lab 5.2 for how to add and delete collision boundaries. Keep the main avatar static for now to test the collision boundary.](#)

7. Make the collision boundary move automatically as the main avatar is being dragged.
 - a. Whilst the main avatar is being dragged, i.e. within touchesMoved function, it needs to delegate the main view to do the following actions:
 - i. Remove all existing collision boundaries for obstacle cars;
 - ii. Add the current car frame to the collision boundary.
 - b. Read Lab 4 for how to use delegate.
8. Set a time-out for the game. **Please make sure each game is no longer than 20 seconds.**
 - a. Read Lab 6 for generating delay.
9. At the end of time-out, show a Game-Over screen. Enable replay.
 - a. Read Lab 3.1 for how to make a view show/hidden.
10. Keep score, which is gained for bypassing an obstacle and lost for colliding.
 - a. Read Lab 5.2 for a solution to create actions after collision/intersection. This code should be put in the same delegate as step 6.
 - b. However, make sure the intersection is not caused by the road view, game-out view, or itself. Read Lab 3 for how to search all subviews so as to exclude these possibilities.
11. Repeat Step 3 –7 to implement coin objects.
12. Further improve and polish the app. Make it more real. Enhance user experience.
 - a. Use Lab 2 to practice Agile management and version control. Remember – always back up your work on GitHub before you leave the desk.

MODULE LEARNING OUTCOMES ASSESSED BY THIS ASSIGNMENT:

a. Knowledge and Understanding:

Successful students will typically have a knowledge and understanding of:

- [2] Principles of mobile operation and usability
- [3] Development and evaluation practices in mobile development

b. Skills and Attributes:

Successful Students will typically be able to:

- [4] Write an app using a well-supported mobile platform and development environment
- [5] Handle issues of connectivity, user experience, accelerometry and location awareness in mobile programming
- [6] Critically evaluate the usability of a mobile app

SUBMISSION REQUIREMENTS:

This assignment is to be submitted and marked anonymously. Students should ONLY use their student ID number to identify themselves on their work. Work submitted via StudyNet for anonymous marking will automatically have an anonymity number allocated to it.

Canvas:

- **Demo Video:** Please read *Canvas/Units/Assignment/Video Recording* for guidance of recording videos.
- **Source Code:** Zip your Xcode project into one file and submit it via Studynet.
- **Cover Page:** Fill in the front page of this Assignment Briefing Sheet and submit via Studynet.
- **Report:** You need to submit a written report containing the self-evaluation of the app you designed, reflecting the usability and future improvements. The word limit is 800 words

excluding references, and marginal excess is permitted. Please note this is a usability report, not a user manual. [Refer to Lecture 6 for usability analysis.](#)

- **ReadMe file:** Optionally, you may wish to submit a ReadMe text file. If you have any special features that cannot be tested using iOS simulator, or if you have put your demo video somewhere else other than Office 365, make notes of them in the ReadMe file to let us know.
- **Links for GitHub and Trello pages.** You can either write comments in the submission or write them in the ReadMe file.

Please use your student ID to name all your submissions required below.

GitHub: Version control is compulsory for this coursework. Your repository on GitHub must be private ([please refer to Tutorial 2 for how to create private repository](#)). Please add b.ip@herts.ac.uk as a member or collaborator of your project. The name of your repository should be your **student ID**.

Trello: Agile management is also compulsory for this coursework. Please invite b.ip@herts.ac.uk to your Kanban board on Trello. The name of your board should be your **student ID**.

Your student ID will be used as an index to identify your work among over 100 submissions. Failure to name your work with your student ID may result in a non-submission mark.

Viva: This is optional. You may be called to attend a viva session if there are reasons to question your code.

FEEDBACK FROM THIS ASSIGNMENT

Each individual student will receive a written feedback about this assignment, as well as marks awarded based on the criteria set in the next section.

MARKS AWARDED FOR:

Components	Marks
Functionalities and Reliability	60%
Development Practice	10%
Usability Practice	20%
Usability Analysis	10%
Total	100%

- **Functionalities and Reliability:** the programming exercise to achieve the required functionalities; reliability refers to the smooth and error-free running of the app
- **Development Practice:** agile management; repository with version control
- **Usability Practice:** friendly user interface for the targeted user group; a highly-usable app for any iPhone models
- **Usability Analysis:** self-reflection of the app design and implementation; discussion of future improvement

Please see next page for the detailed grading criteria of this coursework.

Grading Criteria

	Functionalities and Reliability (60%)	Development Practice (10%)	Usability Practice (20%)	Usability Analysis (10%)
1st	Outstanding use of appropriate technologies. Consistent and accurate execution. Full functions as required, with steps beyond expectations using sophisticated solutions. The app can run smoothly. No crashes or errors after repeated testing.	Effective, frequent, and sustained use of management skills. Well-planned task cards on the Kanban. Version control is efficiently and effectively used. Agile management helps with the project. Regular management behaviours are evident.	Visual impact. Flawless operations. Model compatibility. Proportionally sized displays.	Critical analysis. Evidence-based research. In-depth evaluation. Statements supported by evidence. Sensible suggestions for future improvements. Background research effort indicated by solid reference list. Proper academic writing and report formatting.
2:1	Appropriate use of technologies. Minor errors in technique and/or application with little impact on deliverables. Required components are functional with accuracy. The app can run smoothly for the first or second time, but there may be some errors or crashes after repeated testing.	Good efforts, with occasionally minor setback. Well managed project on regular basis overall. May be lack of a few activities here or there.	Attempt on the visual impact. User-friendly and compatible. Minor setback on display. May work well on certain screens but not others.	Effective analysis. Can benefit from minor tweaks. Commendable quality of work containing all relevant analysis and discussions. May be lack of critical analysis. A minor effort of improvements still required.
2:2	Appropriate use of technologies to the problem domain. Key functionalities implemented. Execution with occasional errors, giving minor impact on intended operations. Occasional crashes.	Some attempt, but needs major improvement. Relatively large time gap found.	Lack of visual impact. Relatively easy to use. Minor compatibility and display issues.	Shortage of research efforts. Satisfying writing overall. Satisfactory level of understanding. Proper concepts are demonstrated with evidence.
3 rd	Genuine attempt on the technologies. Required functions are not partially completed. Delivery of a working product, but with notable errors that require major improvements.	Claimed attempt with insufficient evidence. Only a couple of backup and project management attempts during the whole development process. Mismatched or doubtful submissions made.	No effort on the appeal. Not intuitive to operate. Inadequate visual display.	Verbose and redundant. Lack of concrete content. Brief discussion on the subjects. Familiarity with the usability concept, but unable to form an evaluation.
Fail	Limited use of technologies. A non-working product. Fatal errors. Missing core components. Significant improvements are necessary. Academic misconduct.	Low or little attempt on the management practice. No exercise towards agile management and version control. Fabrication of the process.	Major problems with visual display and compatibility issues.	Inadequate content. Very brief text on the subject. Academic dishonesty.

DEADLINES AND ASSIGNMENT WEIGHTINGS

- | | | | |
|---|---|------------|--|
| 1 | This assignment is worth | 90% | of the overall assessment for this module. |
| 2 | You are expected to spend about | 80 | Hours to complete this assignment to a satisfactory standard |
| 3 | Date assignment set | 02/11/2018 | Date completed assignment to be handed in |
| | | | 17:00 GMT, 07/01/2019, Monday |
| 4 | Target date for return of marked assignment | 04/02/2019 | |

INTERNAL MODERATION

<p>This assignment has been internally moderated.</p> <p>I confirm:</p> <ul style="list-style-type: none"> • That the assignment set, meets the requirements of the module and that the brief provides appropriate content for students to successfully complete the assignment. • That the assessment is at an appropriate level and matches QAA level descriptors and is an appropriate form of assessment within the total range of assessments for this module. • That the marking scheme is attached and that students can determine how marks are allocated. • That this assessment can be completed and marked within University timeframes, and provides detailed feedback (more than just a grade) that supports learning. 	<p>Moderator name, signature and date</p> <p>Wei Ji 16/09/2018</p>
--	---