Entertainment Systems – 1st Term Assignment

This assignment runs over three weeks.
Set date: 20/11/2006
Hand-In date: 11/12/2006

The overall weighting of this assignment is 30% in terms of ES assignment marks (*or 15% of the overall ES mark*).

Your task is to write a simple game. For this you may chose from one of the five options given below (*these will be explained in more detail on separate pages*):

1. Lunar Rescue
2. Space Taxi
3. Scrolling Shooter
4. Asteroids
5. Breakout

Deliverables - You have to provide:

a) One executable file (*and all resource files like graphics, & sounds etc. required for running the program*)
b) All source code files (*commented*) needed for compiling/running the project
c) Post-Mortem-Style Project Report (*1000 words, i.e. up to ca. 2-3 A4 pages*) – see **http://www.gdmag.com/postmort.htm**
d) A short (*HTML*) user manual explaining the workings of your game

Marking Scheme:

Minimum features	50%
Report	25%
Other Features*	25%

* program design, code structure, content (*including user interface*), extra features or other innovative content in terms of graphics (*design*) or additional in-game options (*power-ups etc.*), etc.

Assignment Option A: Lunar Rescue

Your task is to write a simple game resembling the classic lunar rescue arcade game. Lunar rescue involves dropping onto a moon from a mothership, dodging asteroids on the way down to rescue stranded astronauts from the surface. On returning to the mothership the player must avoid being shot down by aliens. Your program must be capable of the following (*minimum features*):

1. Collision detection and resolution - your space craft must be able to collide with the environment and obstacles
2. A user interface that is operated by the keyboard.
3. Use of 2D sprites for in-game graphics
4. Basic gravity
5. Enemies or asteroids that hinder the player (*i.e. they are obstacles*)

Extension ideas:

- Multiple levels
- A basic AI, to make opponents hunt / shoot at the player
- Power-ups
- Arm the player, so they can shoot at enemies, and add shields to protect them (*these could be power-ups*)
- Sound

References:

http://www.klov.com/game_detail.php?letter=L&game_id=8466
http://www.caiman.us/scripts/fw/f2511.html

Assignment Option B: Space Taxi

Your task is to write a simple game resembling the classic space taxi arcade game. Space taxi involves navigating a side-view environment to pick up passengers from platforms and delivering them to other platforms to unlock the exit of the level/game. Your program must be capable of the following (*minimum features*):

1. Collision detection and resolution - your space craft must be able to collide with the environment and obstacles
2. A user interface that is operated by the keyboard.
3. Use of 2D sprites for in-game graphics
4. Random passenger positioning
5. At least one level

Extension ideas:

- Gravity
- Fuel Management (*fuel runs out & must be refilled*)
- Traffic (*other space craft/moving obstacles*)
- Additional levels
- Sound

References

http://www.games-soft.com/Space_Taxi_2.html
http://en.wikipedia.org/wiki/Space_Taxi

Assignment Option C: Simple Scrolling Shooter

Your task is to write a simple game resembling a classic horizontally scrolling shooter (*or shoot'em up*) arcade game. This usually involves some sort of flying vehicle which must dodge incoming enemies and missiles and destroying these enemies. Your program must be capable of the following (*minimum features*):

1. Collision detection and resolution - your space craft must be able to collide with the environment and obstacles, and be shot by enemy craft
2. A user interface that is operated by the keyboard or mouse
3. Use of 2D sprites for in-game graphics
4. At least two types of differently behaving enemies (*i.e. basic enemy AI*)
5. At least one level

Extension ideas:

- Additional obstacles that the player must avoid
- Power-ups
- A HUD
- Multiple levels
- A scrolling background
- Sound

References:

http://www.poke53280.de/artikel/artikel.php?id=31
http://ff2.curvedinfinity.com/

Assignment Option D: Asteroids

Your task is to write a simple game resembling the classic asteroids game. This involves the player steering his space craft to avoid destructable asteroids that cross the playing area – if an asteroid is shot it breaks up into several smaller asteroids. Your program must be capable of the following (*minimum features*):

1. Collision detection and resolution - your space craft must be able to collide with the asteroids
2. A user interface that is operated by the keyboard or mouse
3. Use of 2D sprites for in-game graphics (*SDL*) or geometric shapes (*Java2D*)
4. At least two sizes of asteroid (*i.e. big asteroid must split into smaller asteroids*)
5. The player must have at least two lives to start with

Extension ideas:

- Enemy space ships that shoot at the player
- Power-ups (*e.g. shields, extra lives, teleporter*)
- Additional asteroid sizes
- Sound

References:
http://software.swordfighter.co.uk/asteroids.zip
http://software.swordfighter.co.uk/XSetup.exe

Assignment Option E: Breakout

Your task is to write a simple game resembling the classic breakout arcade game. Breakout (*or Arkanoid*) involves the player destroying a wall of bricks using a bouncing ball which must be prevented from leaving the play area by a player-controlled horizontally moving paddle. Your program must be capable of the following (*minimum features*):

1. Collision detection and resolution - you must provide a paddle, blocks and a ball that bounces off the walls, paddle and blocks
2. A user interface that is operated by the keyboard or mouse
3. Use of 2D sprites for in-game graphics
4. At least one level
5. Multiple block types (*at least: regular, hit twice, indestructible*)

Extension ideas:

- Power-ups (*e.g. wider paddle, faster/slower ball, extra lives*)
- Additional levels (*possibly stored in a file or multiple files*)
- Sound

References:

http://www.arkanoid.com/
http://en.wikipedia.org/wiki/Breakout

guidance notes for project preparation

Acceptable development platforms (*and programming languages*) are MS Windows & Linux and Java, C & C++. Acceptable interfaces/libraries are Java2D and SDL.

When working on a programming project you may come across a problem which has already been solved by somebody other than you. In that case it is not necessary for you to "reinvent the wheel". However all work included in your project (*including assets like graphics and sounds*) that was not produced by yourself must be properly acknowledged in your report and credited in the program source code (*the sections of code that you use must be marked by comments within your project source code*). You are not marked on re-used code but only on original code written by you!

Failure to properly credit your sources is a serious assessment offence and will be handled accordingly. This is not limited to the project source code but also extends to the documentation/report that goes with it:

Plagiarism: the copying or misappropriation of ideas (or their expression), text, software or data (or some combination thereof) without permission and due acknowledgement, i.e. the representation of another person's work as one's own or the use of another person's work without acknowledgement, e.g.

· the direct importation into one's work of more than a single phrase from another person's work without the use of quotation marks and identification of the source
· making a copy of all or part of another person's work and presenting it as one's own by failing to disclose the source
· making extensive use of another person's work, either by summarizing or paraphrasing it merely by changing a few words or altering the order of presentation, without acknowledgement
· the use of the ideas of another person without acknowledgement of the source, or the submission or presentation of work as one's own which is substantially the ideas or intellectual data of another

Other assessment offences relating to projects are:

Collusion: the representation of a piece of unauthorized group work as the work of a single student.

Commissioning another person to complete an assignment which is then submitted as the student's own work.

Computer fraud: the use of the material which belongs to another person and which is stored on a hard or floppy disk without acknowledgement and or without the written permission of the owner.

Duplication: the inclusion in coursework of any material which is identical or substantially similar to material which has already been submitted for any other assessment within the university or elsewhere (for example, the use of essay banks).

Academic Fraud: deliberate deception (*covers many of the above*).

The following URL provides information on how to reference other people's work:
http://www.bournemouth.ac.uk/library/using/guide_to_citing_internet_sourc.html