



Term 1 Assignment



Dates etc.

- project brief set: 20/11/2006
- project tutorials: in the workshop/tutorial slots
- project deadline: 11/12/2006, 12 noon

Assignment Weighting:

30% of coursework mark (*15% of overall ES mark*)

Project Option A: Lunar Rescue

Your task is to write a simple game resembling the classic lunar rescue arcade game. Your program must be capable of the following:

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

Project Option A

For inspiration check out:

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

- start with a simple lunar lander
- then add the return to the mother ship
- only once the basic gameplay has been implemented should graphics and additional functionality (*asteroids, aliens*) be considered



Project Option B: Space Taxi

Your task is to write a simple game resembling the classic space taxi arcade game. Your program must be capable of the following:

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

Project Option B

For inspiration check out:

http://www.games-soft.com/Space_Taxi_2.html

http://en.wikipedia.org/wiki/Space_Taxi

- similar to lunar lander in many aspects of code
- start with 1 platform, then expand
- only once the basic gameplay has been implemented should graphics and additional functionality be considered



Project Option C: A simple Scrolling Shooter

Your task is to write a simple game resembling a classic horizontally scrolling shooter arcade game. Your program must be capable of the following:

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

Project Option C

For inspiration check out:

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

- start with a simple side scrolling ship
- then add the enemies coming in from the side
- only once the basic gameplay has been implemented should graphics and additional functionality (*different enemies, background*) be considered



Project Option D: Asteroids

Your task is to write a simple game resembling the classic asteroids game. Your program must be capable of the following:

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

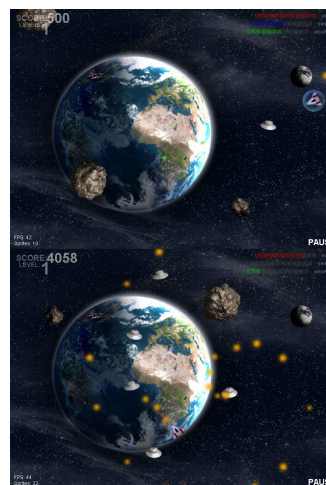
Project Option D

For inspiration check out:

<http://software.swordfighter.co.uk/asteroids.zip>

<http://software.swordfighter.co.uk/XSetup.exe>

- asteroids are best stored in a linked list
- only once the basic gameplay has been implemented should graphics and additional functionality be considered



Project Option E: Breakout (*Arkanoid*)

Your task is to write a simple game resembling the classic breakout game. Your program must be capable of the following:

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

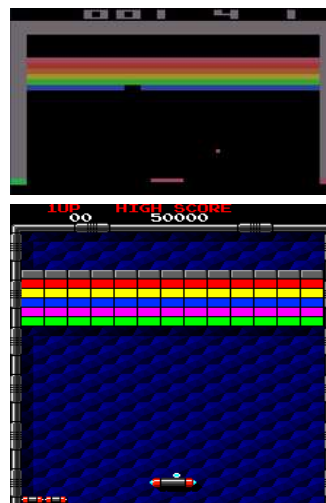
Project Option E

For inspiration check out:

<http://www.arkanoid.com/>

<http://en.wikipedia.org/wiki/Breakout>

- start with a simple pong-like game
- then add bricks
- only once the basic gameplay has been implemented should graphics and additional functionality be considered





Deliverables

- One executable file & all necessary resource files (*images, sounds etc.*).
- All source code files (*commented*) needed for compiling the project.
- Post-Mortem-Style Project Report (*up to ca. 2-3 A4 pages – about 1000 words*)
see: <http://www.gdmag.com/postmort.htm>
- Short (*HTML*) User Manual for your game.



marking scheme

minimum features [€] -	50%
report -	25%
other marks* -	25%

[€] note that the more of you attempt the same project, the expectation regarding the outcome will be higher since you will have access to a larger knowledge base – i.e.. chances of getting a good mark increase if you choose a less popular project

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



guidance notes

- 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*).



guidance notes

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 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 summarising 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



guidance notes

Other assessment offences relating to projects are:

- **Collusion:** the representation of a piece of unauthorised 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).

This URL provides information on how to reference other people's work:

http://www.bournemouth.ac.uk/library/using/guide_to_citing_internet_sourc.html