

SESSION: 2018/19 DIET 1

Module Title: Games Programming 3

Module Code: MHI622946

Level: 4

Module Leader: Dr. Richard Holden

Individual Marks Summary:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Component | A: Environment set-up | B: Game Coding | C: Persistence | D: Report |
| Component % | 5 | 40 | 15 | 40 |

|  |  |  |
| --- | --- | --- |
| Student Name | Student ID | Student E-mail |
|  |  |  |

Student Declaration

This piece of work is not plagiarized. It is my own original work and has not been submitted elsewhere in fulfilment of the requirements of this or any other award.

Signature: Date:

Contents

No table of contents entries found.

# Moore’s Law

Moore’s Law (ML) is the observation that the transistor quantity and performance of integrated circuits doubles every year (Moore, 1995; Schaller, 1997; Thompson e Parthasarathy, 2006; Mack, 2011).

# Multi-Core Technology

Multi-Core technology represents innovative advancements such as multi-core processors, which are defined as a singular processor containing two or more execution cores. As stated by (Gepner e Kowalik, 2006), multi-core technology is an important innovation as it drives forward multi-threading and parallelism.

## Parallelism

Parallelism involves the execution of multiple tasks in a computer system simultaneously, this works in contradiction to sequential programming, which incorporates the execution of tasks in an arranged order. Despite modern computers now shipping with multiple processors, often the system cannot fully take advantage of the hardware due to restrictions in the design of software. As presented by (Olszewski *et al.*, 2009) , multithreaded programs which provide shared memory provide interweaved accesses to shared data. (Ranger *et al.*, 2007) highlights the need for software to be programmed parallelly to exploit multi-core processors, as they become more common.

# Multi-Threading

Multi-threading is a computation model which allows the existence of many threads to execute a single process. This can help to break singular tasks down into smaller subtasks, which is independently executed by each separate thread. Although threads may share resources of the process, referencing data which is being altered by another thread can cause issues. Therefore, clear design of the software’s structure critical.

# References

GEPNER, P.; KOWALIK, M. F. Multi-core processors: New way to achieve high system performance. Parallel Computing in Electrical Engineering, 2006. PAR ELEC 2006. International Symposium on, 2006, IEEE. p.9-13.

MACK, C. A. Fifty years of Moore's law. **IEEE Transactions on semiconductor manufacturing,** v. 24, n. 2, p. 202-207, 2011. ISSN 0894-6507.

MOORE, G. E. Lithography and the future of Moore's law. Integrated Circuit Metrology, Inspection, and Process Control IX, 1995, International Society for Optics and Photonics. p.2-18.

OLSZEWSKI, M.; ANSEL, J.; AMARASINGHE, S. Kendo: efficient deterministic multithreading in software. **ACM Sigplan Notices,** v. 44, n. 3, p. 97-108, 2009. ISSN 1605584061.

RANGER, C. et al. Evaluating mapreduce for multi-core and multiprocessor systems. High Performance Computer Architecture, 2007. HPCA 2007. IEEE 13th International Symposium on, 2007, Ieee. p.13-24.

SCHALLER, R. R. Moore's law: past, present and future. **IEEE spectrum,** v. 34, n. 6, p. 52-59, 1997. ISSN 0018-9235.

THOMPSON, S. E.; PARTHASARATHY, S. Moore's law: the future of Si microelectronics. **Materials today,** v. 9, n. 6, p. 20-25, 2006. ISSN 1369-7021.