



# Lecture 11 – The Package Software Industry

Karl R. Wilcox  
K.R.Wilcox@reading.ac.uk



# Objectives

- To understand the different roles of organisations in the package software industry
- We will consider the games industry
  - Others are similar
  - Probably less complex!
- Today's practical
  - Investigating Software Licenses



# Structure of the Industry

- **Developers**
- **Publishers**
- **Distributors**
- **Retailers**
- **Hardware Manufacturers**
  - (Mostly only relevant to Games Industry)



# Publishers

- **Bring products to the market**
- **Create “schedules” of planned releases**
- **Responsible for**
  - Commissioning & funding products from developers
  - Marketing & promotion
  - Organising manufacture
  - Obtaining agreements with distributors
- **Make money from**
  - Royalties from sales
    - To cover advances to developers
    - Shared royalties thereafter



# Developers

- **Create the product**
- **By employing**
  - Programmers, artists, designers, sound engineers, musicians, actors, producers, writers...
- **Responsible for**
  - Writing, testing & supporting the product
  - Finding a publisher
- **Make money from**
  - Advances from publishers
  - Royalties from successful products



# Distributors

- **Maintain stocks of product**
- **Sell to “lower level” distributors or to retailers**
- **May have exclusive agreements with publishers**
- **Responsible for**
  - Logistics ( delivering product as required)
  - Obtaining agreements with publishers and retailers
- **Make money from**
  - Selling products for more than they bought them



# Retailers

- **Sell directly to the public**
- **May be High Street / Mail Order / Internet**
- **Responsible for**
  - Obtaining agreements with distributors
  - May run their own promotions
- **Makes money from**
  - Selling games for more than they paid the distributor



# Hardware Manufacturers

- Only have a significant role in console games
- Validate games for use on console
- Responsible for
  - Development & manufacture of console hardware
  - Marketing of the console “platform”
- Makes money from
  - Licenses to develop software for that console
  - Royalties from software sales
  - (Usually lose money from sales of console hardware)





# Product Types - Licensed

- **Based on existing Intellectual Property**
  - e.g. film / TV programme / book
- **Licenses usually acquired by publisher who then seeks developer**
- **Sometimes developers seek rights directly**
- **Advantages**
  - Recognition factor / joint marketing
  - Therefore increased sales
- **Disadvantages**
  - Very large up front costs for rights + royalties
  - Timescales may be driven by film release dates etc.



# Product Types - Original

- **Usually created by developers**
  - Idea then “sold” to publishers to fund further development
  - Banks & other backers may fund early development
- **Advantages**
  - No licensing or additional royalty costs
  - Control of timescales & other aspects
- **Disadvantages**
  - Riskier (especially for developer in early stages)
  - Product must stand on its own merit



# Product Types - Conversion

- **Simple conversion from one format to another**
  - PC to Mac (or vice versa)
  - PC to Linux (or v. v.)
  - For games only – between consoles
- **Some products developed for multiple platforms in parallel**
- **Advantages**
  - Proven demand, tested product
- **Disadvantages**
  - Increased support required
  - Code / features may be difficult to port
    - Especially between games consoles & PCs



# Developing for consoles

- **Additional up-front costs for development licence & tools**
- **Ongoing royalty payments to console manufacturers**
- **Must submit product to manufacturers for testing / validation**
- **Closed system**
  - Defined, stable platform
  - Minimal ongoing support costs
  - Extreme difficulty in fixing bugs after release!



# Developing for PCs

- No licensing costs for use of platform
- Cheap (or free) development tools
- No royalty to hardware manufacturers
- Open platform
- Extreme variability in operating environment
- Expect high, ongoing support costs
- Possible to issue patches / updates
  - Get it right 2<sup>nd</sup> time (or 3<sup>rd</sup>, 4<sup>th</sup>, 5<sup>th</sup> ...)



# Complications

- **A single company may take on more than one role**
  - Especially Publisher & Developer
  - Also hardware manufacturer and publisher
  - But still deal with others of the same type
- **Distribution rights and licensing rights may have geographical restrictions**
  - Different publishers / distributors in different parts of the world



# Differences Between Games & Other Software Products

- **Not so much licensing for other software**
  - Perhaps “Alan Titchmarsh Garden Designer”
- **Very little non-game software for consoles**
  - DVD players for Xbox / Playstation 2
- **Need for fewer “creative” staff**
  - Artists, musicians, actors
- **Simpler testing regime**
  - Just functionality, no real equivalent of “Gameplay”
- **Lower costs of entry / development permit other funding models**
  - See next slide



# Other Funding Models - 1

- **Freeware**
  - Typically reduced functionality
  - Designed to raise awareness of the full featured product, or of the developer
- **Open Source Software**
  - Freely available at source code level
  - Changes must be made equally available
  - Developed by “community”
  - Or by commercial organisations
  - Possible to make money from support / training / consultancy / repackaging & distribution





# Other Funding Models - 2

- **Shareware**
  - Developer writes & distributes own software via internet
  - Product can be freely distributed but normally limited in some way
    - Time restriction
    - File size restriction
    - Advertisement insertion
  - User pays small amount to developer to unlock full feature set
- **Adware**
  - Free, full function software but user must view advertisements
    - E.g. Opera Browser, Eudora Mail client



# Today's Practical

- Investigating Software Licensing
- Understanding End User Licence Agreements
  - Or trying to!
- **REMEMBER TO SIGN OFF ON THE REGISTRATION SHEET!**