Practical 2 Part 1

# Task 1: Game Engines 9 (AA1) – 7 marks

## 2D Game Engines

1. LÖVE
   1. Its mobile targets are both Android and iOS
   2. Its desktop targets are Windows, OSX, Linux and HTML5 (unofficially)
   3. Its development platforms are Windows Vista+, macOS 10.7+, Linux and Android
   4. Languages which can be used are Lua and C++
   5. Its free to make use of
2. libGDX
   1. Its mobile targets are iOS, Android, Blackberry, Desktop and HTML5
   2. Its desktop targets are Windows, OSX, Linux and HTML5
   3. Its development platforms are Windows, OSX and Linux
   4. The programming language used is Java
   5. Its free to make use of
3. Construct 2
   1. Its mobile targets are IOS and Windows
   2. Its desktop targets are Linux, Windows and Macintosh
   3. Its development platform is Windows
   4. The programming language used is JavaScript
   5. The limited version is free and the full version is €159.99
4. Oxygine 2D C++ Game Framework
   1. Its mobile target are Android, iOS and Web
   2. Its desktop targets are Windows, OSX and Linux
   3. Its development platforms are Windows, OSX and Linux
   4. The programming language used is C++
   5. Its free to make use of

Reference: <https://www.slant.co/topics/341/~best-2d-game-engines>

## 3D Game Engines

1. Godot
   1. Its mobile targets are iOS, Android and BB10
   2. Its desktop targets are Windows, OSX, Linux and HTML5
   3. Its development platforms are Windows, OSX and Linux
   4. The programming language used are C++(library), C# and GDscripts
   5. Its free to make use of
2. Unreal Engine 4
   1. Its mobile targets are iOS and Android
   2. Its desktop targets are Windows, Mac OS X, Linux, StreamOs and HTML5
   3. Its development platforms are Windows, Mac OS X and Linux
   4. The programming language used are C++ and Blueprints (Visual Scripting)
   5. Its free to make use of
3. Unity
   1. Its mobile targets are Windows, iOS, Android, BlackBerry 10 and Tizen
   2. Its desktop targets are Windows, OSX and Linux
   3. Its development platforms are Windows, OSX and Linux
   4. Its console targets are Xbox 360, Xbox One, Wii U, PlayStation 3, PlayStation 4, PlayStation Vita, Nintendo Switch and Nintendo 3DS
   5. Its free to make use of
4. CryEngine
   1. Its mobile targets are iOS and Android
   2. Its desktop targets are Windows and Linux
   3. Its development platform is Windows
   4. The programming language used C++, C# and Lua
   5. Its free to make use of

Reference: <https://www.slant.co/topics/8080/~3d-game-engines-for-beginners>

# Task 2: Gaming Programing Languages (AA2) – 7 marks

1. C++
   1. It’s and Object Oriented language
   2. Virtual Functions
2. Lua
   1. Syntax
   2. Control Flow
3. JavaScript
   1. Universal Support
   2. Structured
4. Python
   1. Multi-paradigm programming language
   2. Dynamic Name Resolution (late binding)
5. C
   1. Inline Functions
   2. Multiple Distinct Memory Banks
6. Objective-C
   1. Pre-processing
   2. Categories
7. AngelScript
   1. Single-Inheritance
   2. Object Handles

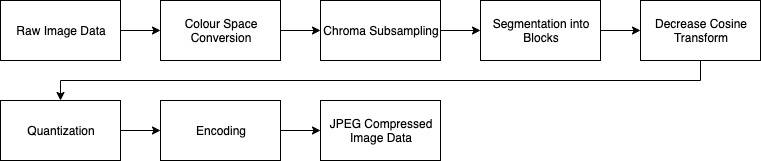
Reference: <https://en.wikipedia.org/wiki/List_of_game_engines>

# Task 3: Compression in multimedia (KU4) – 5 marks

## The importance of compression in multimedia objects (100 words)

It is important to compress multimedia objects as this way less space is taken up by these files. For optimal efficiency when managing large multimedia data objects, these files must be compressed to eliminate redundancies in the pattern of the data. An easy example of this is when you have a black pixel followed by 15 white consecutive pixels. Storing the code for each white pixel makes to file size for storing these objects unnecessarily large. However, when a coding mechanism is used, only the count of theses 15 white pixels is stored. This removes any redundancy whilst making the data objects smaller. Thus, transmission of these multimedia objects over the network will be much quicker which results in a reduction of storage space and transmission costs.

## Explain in detail using diagrams how compression in an image works



**Raw Image Data:**

This is the original picture in its most storage space consuming form

**Color Space Conversion:**

The RGB (Red, Green, Blue) data of the image is converted to YCbCr (Luminance, Chrominance Blue, Chrominance Red) color channels. The YcbCr format is made up of three interdependent channels which is processed without color conversion.

**Chroma Subsampling:**

**Segmentation into Blocks:**

**Decrease Cosine Transform:**

**Quantization:**

**Encoding:**

**JPEG Compressed Image Data:**

This is the image after all the compression stages where done