**Histoire du chip 8**

Le chip 8 est un langage de programmation hexadécimal ayant été inventé dans le but de simplifier la création de jeux pour des systèmes 8 bits. Ce langage utilise une machine virtuelle, ce qui permet aux programmes chip 8 d’être portable dans la mesure où une machine virtuelle existe pour un système donné. Des interpréteurs chip 8 ont été conçu pour de nombreux systèmes : pour DOS, Windows, mac osx, linux, mais aussi pour la calculatrice graphique HP-48 et même en version site web.

**Qu’est-ce qu’un émulateur?**

**Qu’est-ce qu’une machine virtuelle?**

**Comment fonctionne un programme chip 8?**

Un programme chip 8, ou tout autre programme compilé en général, est composé d’instructions de base spécifique au processeur qui doit les lires. Les instructions peuvent être de taille fixe, comme pour la chip 8 ou l’architecture de processeur ARM, ou de taille variable, comme l’architecture de processeur x86. Chaque instruction est composée de l’opcode et de champs servant de paramètres à l’opcode. Ainsi, l’opcode représente la nature de l’instruction alors que les champs permettent de déterminer d’autres informations pertinentes en lien avec l’opcode. Par exemple, pour la chip 8, dont les instructions ont une taille de 16 bits, l’instruction 0x1345 (souvent représenté comme étant 0x1NNN) se décompose ainsi : 0x1 (le opcode) 345 (ou NNN, le champ).

La signification et la longueur des champs dépend de l’opcode. La plupart du temps, dans la documentation du processeur ou de la machine virtuelle les opcodes sont représentés avec des symboles à la place des champs. Par exemple, pour la chip 8, les symboles suivants sont utilisés : NNN (adresse), NN (constante 8 bits), N (constantes 4 bits), X (identifiant de registre 4 bits), Y (identifiant de registre 4 bits).