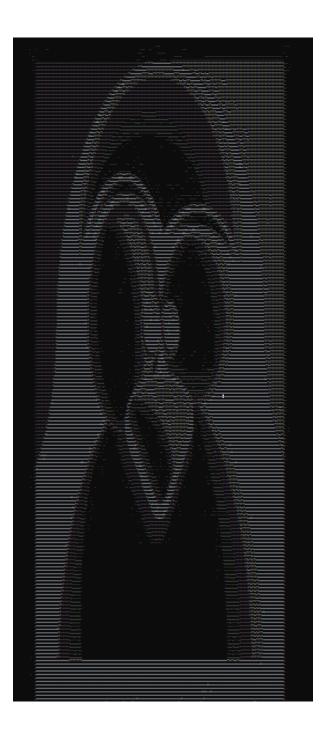
SAE S1.01 Implémentation d'un besoin client

Version 1 La base

```
#include <iostream>
                                                             for(size t i = 0; i < E.size(); i++)</pre>
#ifdef WIN32
#endif //
                                                                  std::cout << E[i] << «\n»;</pre>
#include <Windows.h>
#include <fstream>
                                                             for (size_t j = 0; j < hauteur; j++)</pre>
#include <sstream>
#include <vector>
                                                                  std::vector<char> donnees(largeur);
#include <string>
                                                                  image.read(donnees.data(), largeur);
                                                                 for (size_t k = 0; k < donnees.size(); k++)</pre>
int main()
                                                                      unsigned int don = donnees[k];
#ifdef _WIN32
                                                                      while (don > 255)
    SetConsoleCP(CP UTF8);
    SetConsoleOutputCP(CP_UTF8);
                                                                          don = don \% 256;
#endif // _WIN32
                                                                      if (don >= 0 \&\& don < 32)
    std::string fichier;
                                                                          std::cout << «W»;</pre>
    std::cout << «Entrez le nom du fichier pgm : «;</pre>
                                                                      if (don > 31 && don < 64)
    std::cin >> fichier;
                                                                          std::cout << «w»;</pre>
    std::ifstream image(fichier,
                                                                      if (don > 63 && don < 96)
                         std::ios base::binary);
                                                                          std::cout << «l»;</pre>
    std::vector<std::string> E;
                                                                      if (don > 95 && don < 128)
    std::string entete;
                                                                          std::cout << «i»;</pre>
    unsigned int largeur = 0;
                                                                      if (don > 127 && don < 160)
    unsigned int hauteur = 0;
                                                                          std::cout << «:»;</pre>
                                                                      if (don > 159 && don < 192)
    if (!image.is open())
                                                                          std::cout << «,»;</pre>
        std::cerr << «Problème d'ouverture du
                                                                      if (don > 191 && don < 224)
                      fichier.\n»;
                                                                          std::cout << «.»;</pre>
    }
                                                                      if (don > 223 && don < 256)
    else
                                                                          std::cout << « «;
        for (size t i = 0; i < 4; i++)
                                                                  std::cout << '\n';</pre>
            std::getline(image, entete);
            E.push back(entete);
            if (i == 2)
                                                         Resultat
                                                         Entrez le nom du fichier pgm : linuxmancho.pgm
                 std::stringstream sstr(entete);
                 sstr >> largeur >> hauteur;
                                                         # Converted from linuxmancho.jpg by img2pgm.
                                                         256 256
                                                         255
```



Version 2 Enregistrement de l'Ascii Art

```
#include <iostream>
                                                            for(size t i = 0; i < E.size(); i++)</pre>
#ifdef WIN32
#endif //
#include <Windows.h>
                                                                std::cout << E[i] << «\n»;
#include <fstream>
                                                            for (size_t j = 0; j < hauteur; j++)</pre>
#include <sstream>
#include <vector>
#include <string>
                                                                std::vector<char> donnees(largeur);
                                                                image.read(donnees.data(), largeur);
                                                                for (size t k = 0; k < donnees.size(); k++)</pre>
int main()
#ifdef _WIN32
                                                                    unsigned int don = donnees[k];
   SetConsoleCP(CP_UTF8);
                                                                    while (don > 255)
    SetConsoleOutputCP(CP_UTF8);
#endif // _WIN32
                                                                        don = don % 256;
    std::string fichier;
                                                                    if (don >= 0 && don < 32)
    std::cout << «Entrez le nom du fichier pgm : »;</pre>
                                                                        out << «W»;
    std::cin >> fichier;
                                                                    if (don > 31 && don < 64)
    std::string fichier_texte;
                                                                        out << «w»;
    std::cout << «Entrez le nom du fichier texte :»;</pre>
                                                                    if (don > 63 && don < 96)
    std::cin >> fichier texte;
                                                                        out << «l»;
    std::ifstream image(fichier, std::ios_
                                                                    if (don > 95 && don < 128)
base::binary);
                                                                        out << «i»;
    std::ofstream out(fichier_texte);
                                                                    if (don > 127 && don < 160)
    std::vector<std::string> E;
                                                                        out << «:»;
    std::string entete;
                                                                    if (don > 159 && don < 192)
    unsigned int largeur = 0;
                                                                        out << «,»;
    unsigned int hauteur = 0;
                                                                    if (don > 191 && don < 224)
    if (!image.is_open())
                                                                        out << «.»;
                                                                    if (don > 223 && don < 256)
        std::cerr << «Problème d'ouverture du
                                                                        out << « »;
fichier.\n»;
                                                                out << '\n';
   }
    else
                                                        }
        for (size t i = 0; i < 4; i++)
            std::getline(image, entete);
            E.push_back(entete);
            if (i == 2)
                std::stringstream sstr(entete);
                sstr >> largeur >> hauteur;
```



Version 3 Choix de la palette

```
console.h
                                                        void conversion(std::string fichier,
                                                                                                                             if (don > 63 && don < 96)
#pragma once
                                                                        std::string fichier texte,
                                                                                                                                 out << tab[2];
#include <string>
                                                                        std::array<std::string, 8> tab)
                                                                                                                             if (don > 95 && don < 128)
                                                                                                                                 out << tab[3];
std::string entree nom fichier();
                                                            std::ifstream image(fichier,
                                                                                                                             if (don > 127 && don < 160)
std::string entree_nom_texte();
                                                                                 std::ios base::binary);
                                                                                                                                 out << tab[4];
int entree choix();
                                                            std::ofstream out(fichier texte);
                                                                                                                             if (don > 159 && don < 192)
void conversion(std::string fichier, std::string
                                                            std::vector<std::string> E;
                                                                                                                                 out << tab[5];
fichier texte, std::array<std::string, 8> tab);
                                                                                                                             if (don > 191 && don < 224)
                                                            std::string entete;
                                                            unsigned int largeur = 0;
                                                                                                                                 out << tab[6];
console.cpp
                                                            unsigned int hauteur = 0;
                                                                                                                             if (don > 223 && don < 256)
#include <iostream>
                                                            if (!image.is_open())
                                                                                                                                 out << tab[7];
#include <fstream>
                                                                std::cerr << «Problème d'ouverture du</pre>
                                                                               fichier.\n»:
                                                                                                                         out << '\n';
#include <vector>
                                                            else
#include <sstream>
#include <string>
                                                                                                                }
#include <array>
                                                                for (size_t i = 0; i < 4; i++)
#include «console.h»
                                                                    std::getline(image, entete);
                                                                    E.push back(entete);
std::string entree nom fichier()
                                                                    if (i == 2)
    std::string fichier;
    std::cout << «Entrez le nom du fichier pgm : »;</pre>
                                                                         std::stringstream sstr(entete);
    std::cin >> fichier;
                                                                         sstr >> largeur >> hauteur;
    return fichier;
std::string entree_nom_texte()
                                                            for (size t i = 0; i < E.size(); i++)
    std::string fichier texte;
    std::cout << «Entrez le nom du fichier texte :»;</pre>
                                                                std::cout << E[i] << «\n»;
    std::cin >> fichier texte;
    return fichier texte;
                                                            for (size t j = 0; j < hauteur; j++)
}
int entree choix()
                                                                std::vector<char> donnees(largeur);
                                                                image.read(donnees.data(), largeur);
{
                                                                for (size t k = 0; k < donnees.size(); k++)</pre>
       int choix;
       std::cout << «Veuillez choisir votre</pre>
palette de couleur \n»
                                                                    unsigned int don = donnees[k];
               «1 : palette\n»
                                                                    while (don > 255)
               «2 : palette2\n»
               «3 : paletteUTF8\n»
                                                                         don = don \% 256;
               «4 : palette2UTF8\n»;
       std::cin >> choix;
                                                                    if (don >= 0 \&\& don < 32)
                                                                         out << tab[0];
       return choix;
                                                                    if (don > 31 && don < 64)
}
                                                                         out << tab[1];
```

```
fonction.h
#pragma once
#include <fstream>
#include <string>
#include <array>
std::array<std::string, 8> palette(int choix);
fonction.cpp
#include <iostream>
#include <fstream>
#include <vector>
#include <string>
#include <array>
#include «fonction.h»
std::array<std::string, 8> palette(int choix)
    std::string couleur;
    std::array<std::string, 8> tab;
    switch (choix)
    {
    case 1:
        std::ifstream palette(«palette.txt»);
        for (size t i = 0; i < tab.size(); i++)</pre>
            std::getline(palette, couleur);
            tab[i] = couleur;
        break;
    case 2:
        std::ifstream palette(«palette2.txt»);
        for (size_t i = 0; i < tab.size(); i++)</pre>
            std::getline(palette, couleur);
            tab[i] = couleur;
        break;
```

```
case 3:
{
    std::ifstream palette(«paletteUTF8.txt»);
    for (size_t i = 0; i < tab.size(); i++)
    {
        std::getline(palette, couleur);
        tab[i] = couleur;
    }
    break;
}
case 4:
{
    std::ifstream palette(«palette2UTF8.txt»);
    for (size_t i = 0; i < tab.size(); i++)
    {
        std::getline(palette, couleur);
        tab[i] = couleur;
    }
    break;
}
preturn tab;</pre>
```



```
solution.cpp
#include <iostream>
#ifdef WIN32
#endif //
#include <Windows.h>
#include <fstream>
#include <vector>
#include <string>
#include <array>
#include «console.h»
#include «fonction.h»
int main()
#ifdef WIN32
    SetConsoleCP(CP UTF8);
    SetConsoleOutputCP(CP_UTF8);
#endif // WIN32
        std::string fichier = entree nom fichier();
    std::string fichier texte = entree nom texte();
    std::ifstream image(fichier, std::ios_
base::binary);
    conversion (fichier, fichier texte,
palette(entree choix()));
```

Version 4 Arguments de la ligne de commande

```
console.h
#pragma once
#include <string>
void conversion(std::string fichier, std::string fichier_texte,
std::vector<std::string> tab);
console.cpp
#include <iostream>
#include <fstream>
#include <vector>
#include <sstream>
#include <string>
#include «console.h»
void conversion(std::string fichier, std::string fichier_texte,
std::vector<std::string> tab)
    std::ifstream image(fichier, std::ios base::binary);
    std::ofstream out(fichier texte);
    std::vector<std::string> E;
    std::string entete;
    unsigned int largeur = 0;
    unsigned int hauteur = 0;
    if (!image.is open())
        std::cerr << «Problème d'ouverture du fichier.\n»;</pre>
    else
        for (size t i = 0; i < 4; i++)
            std::getline(image, entete);
            E.push back(entete);
            if (i == 2)
                std::stringstream sstr(entete);
                sstr >> largeur >> hauteur;
    for (size t i = 0; i < E.size(); i++)
        std::cout << E[i] << «\n»;
```

```
for (size t j = 0; j < hauteur; j++)
        std::vector<char> donnees(largeur);
        image.read(donnees.data(), largeur);
        for (size t k = 0; k < donnees.size(); k++)
            unsigned int don = donnees[k];
            if (don > 255)
                don = don \% 256;
            int intervalle = 256 / tab.size();
            int nb = don / intervalle;
            out << tab[nb];</pre>
        out << '\n';
fonction.h
#pragma once
#include <string>
#include <vector>
std::vector<std::string> palette(std::string nom);
fonction.cpp
#include <iostream>
#include <fstream>
#include <vector>
#include <string>
#include «fonction.h»
std::vector<std::string> palette(std::string nom)
    std::string couleur;
    std::vector<std::string> tab;
    std::ifstream palette(nom);
    while (!palette.eof())
        std::getline(palette, couleur);
        tab.push_back(couleur);
    return tab;
Cette version ne fonctionne pas non plus mais nous ne savons pas
pourquoi.
```

```
Solution.cpp
#include <iostream>
#ifdef WIN32
#endif //
#include <Windows.h>
#include <fstream>
#include <vector>
#include <string>
#include <array>
#include <iomanip>
#include «console.h»
#include «fonction.h»
int main(int argc, char* argv[])
#ifdef WIN32
   SetConsoleCP(CP UTF8);
   SetConsoleOutputCP(CP UTF8);
#endif // WIN32
    std::string fichier;
    std::string fichier texte;
    std::string nom palette = «palette.txt»;
       std::vector<std::string> argument = { «--input», «--output»,
«--palette», «--help» , «--width», «--height» };
       for (int i = 0; i < argc; ++i)
              if (argv[i] == argument[0])
                      if (argv[i + 1] != argument[1])
                              fichier = argv[i + 1];
                      else
                              std::cout << «Veuillez entre le nom de votre</pre>
fichier(.pgm) : \n»;
                              std::cin >> fichier;
              if (argv[i] == argument[1])
                      if (argv[i + 1] != argument[2])
                              fichier texte = argv[i + 1];
                      else
                              std::cout << «Veuillez entre le nom de votre</pre>
fichier texte(.txt) : \n»;
                              std::cin >> fichier texte;
```

```
if (argv[i] == argument[2])
                         if (argv[i + 1] != argument[3])
                                 nom palette = argv[i + 1];
                         else
                                 nom palette = «palette.txt»;
                 if (argv[i] == argument[3])
                         std::cout << «Usage :\nSAE Impl.exe [options]\n\nOptions</pre>
:\n»;
                         std::cout << <--input fichier> << std::setw(15) << < <
                                 << «Spécifie le fichier image à convertir\n»;
                         std::cout << std::setw(30) << « «
                                  << «Si ce paramètre n'est pas spécifié, le fichier
est de-mandé via la console.\n»:
                         std::cout << <--output fichier> << std::setw(14) << « «
                                 << «Spécifie le nom du fichier texte qui
contiendra l'Ascii Art.\n»;
                         std::cout << std::setw(30) << « «
                                 << «Si ce paramètre n'est pas spécifié, l'Ascii
Art est sortie dans la console.\n»;
                         std::cout << «--palette fichier» << std::setw(13) << « «</pre>
                                 << «Spécifie un fichier texte contenant la palette
de cou-leur Ascii.\n»:
                         std::cout << std::setw(30) << « «
                                 << «Chaque ligne du fichier contient un
charactère en UTF - 8, du plus sombre au plus clair.\n»;
                         std::cout << std::setw(30) << « «
                                 << «Si ce paramètre n'est pas spécifié, la
palette par dé-faut est \»W\», \»w\», \»i\», \»:\», \»,\», \».\», \»
\» \n»;
                         std::cout << <--help> << std::setw(24) << « «
                                 << «Affiche cette aide.»;
                         return 0:
    std::ifstream image(fichier, std::ios base::binary);
    conversion (fichier, fichier texte, palette(nom palette));
 AE_Impl.exe [options]
options :
--input fichier
                      Spécifie le fichier image à convertir
                      Si ce paramètre n'est pas spécifié, le fichier est demandé via la console.
 output fichier
                      Spécifie le nom du fichier texte qui contiendra l'Ascii Art.
                      Si ce paramètre n'est pas spécifié, l'Ascii Art est sortie dans la console.
 -palette fichier
                      Spécifie un fichier texte contenant la palette de couleur Ascii.
                      Chaque ligne du fichier contient un charactère en UTF - 8, du plus sombre au plus clair.
                      Si ce paramètre n'est pas spécifié, la palette par défaut est "W", "w", "l", "i", ":".
```

Affiche cette mide.

Version 5 Réduction de la taille des images

```
console.h
#pragma once
                                                           int hmax, lmax;
#include <string>
                                                           if (hauteur max == 1)
                                                               hmax = 1:
                                                                                                               fonction.h
void conversion(std::string fichier, std::string
                                                           else
                                                                                                               #pragma once
fichier texte, std::vector<std::string> tab, int
                                                                                                               #include <string>
                                                               hmax = hauteur / hauteur max;
hauteur max, int largeur max);
                                                           if (largeur max == 1)
                                                                                                               #include <vector>
                                                               lmax = 1;
console.cpp
                                                           else
#include <iostream>
                                                               lmax = largeur / largeur max;
#include <fstream>
                                                                                                               fonction.cpp
                                                           for (size t i = 0; i < E.size(); i++)
#include <vector>
                                                                                                               #include <iostream>
#include <sstream>
                                                                                                               #include <fstream>
                                                                                                               #include <vector>
#include <string>
                                                               std::cout << E[i] << «\n»;
#include «console.h»
                                                                                                               #include <string>
                                                           for (size t i = 0; i < hauteur; i += hmax)</pre>
                                                                                                               #include «fonction.h»
                                                               for (size_t j = 0; j < hmax; j++)
void conversion(std::string fichier, std::string
fichier texte, std::vector<std::string> tab, int
hauteur max, int largeur max)
                                                                    std::vector<char> donnees(largeur);
                                                                                                                   std::string couleur;
                                                                    image.read(donnees.data(), largeur);
    std::ifstream image(fichier, std::ios
                                                                    tablo.push back(donnees);
base::binary);
                                                                                                                   while (!palette.eof())
    std::ofstream out(fichier texte);
                                                       //stockage de h lignes ds un tableau
    std::vector<std::string> E;
                                                               for (size t k = 0; k < largeur; k += lmax)</pre>
    std::string entete;
    std::vector<std::vector<char>> tablo;
    unsigned int largeur = 0;
                                                                    unsigned int moy = 0;
                                                                                                                   return tab;
                                                                    for (size t m = 0; m < hmax; m++)
    unsigned int hauteur = 0;
    if (!image.is open())
                                                                        for (size t n = 0; n < lmax; n++)
                                                                                                               Solution.cpp
        std::cerr << «Problème d'ouverture du
                                                                                                               #include <iostream>
fichier.\n»;
                                                                             moy += (int)tablo[m][n];
                                                                                                               #ifdef WIN32
   }
                                                                                                               #endif //
    else
                                                                                                               #include <Windows.h>
                                                                        mov /= lmax;
                                                                                                               #include <fstream>
        for (size t i = 0; i < 4; i++)
                                                                    moy /= hmax;
                                                                                                               #include <vector>
                                                                    if (mov > 255)
                                                                                                               #include <string>
                                                                        moy = moy \% 256;
                                                                                                               #include <array>
            std::getline(image, entete);
            E.push back(entete);
                                                                    int intervalle = 256 / tab.size();
                                                                                                               #include <iomanip>
            if (i == 2)
                                                       //définition de la répartition plage de couleur
                                                                                                               #include «console.h»
                                                                    int nb = moy / intervalle;
                                                                                                    //plage
                                                                                                               #include «fonction.h»
                std::stringstream sstr(entete);
                                                       de couleur
                sstr >> largeur >> hauteur;
                                                                    out << tab[nb];</pre>
                                                               out << '\n';
```

```
std::vector<std::string> palette(std::string nom);
std::vector<std::string> palette(std::string nom)
    std::vector<std::string> tab;
    std::ifstream palette(nom);
        std::getline(palette, couleur);
        tab.push back(couleur);
```

```
int main(int argc, char* argv[])
#ifdef _WIN32
    SetConsoleCP(CP UTF8);
    SetConsoleOutputCP(CP UTF8);
#endif // WIN32
    std::string fichier;
    std::string fichier texte;
    std::string nom palette = «palette.txt»;
    std::string largeur max;
    std::string hauteur max;
    std::vector<std::string> argument = { «--input», «--output», «--palette»,
«--help» , «--width», «--height» };
       for (int i = 0; i < argc; ++i)
               if (argv[i] == argument[0])
                      if (argv[i + 1] != argument[1])
                              fichier = argv[i + 1];
                      else
                              std::cout << «Veuillez entre le nom de votre</pre>
fichier(.pgm) : \n»;
                              std::cin >> fichier;
                      }
               if (argv[i] == argument[1])
                      if (argv[i + 1] != argument[2])
                              fichier texte = argv[i + 1];
                      else
                              fichier texte = «rien»;
               if (argv[i] == argument[2])
                      if (argv[i + 1] != argument[3])
                              nom palette = argv[i + 1];
               if (argv[i] == argument[3])
                      std::cout << «Usage :\nSAE Impl.exe [options]\n\nOptions</pre>
:\n»;
                      std::cout << «--input fichier» << std::setw(15) << « «</pre>
                              << «Spécifie le fichier image à convertir\n»;
                      std::cout << std::setw(30) << « «
                              << «Si ce paramètre n'est pas spécifié, le fichier
est demandé via la console.\n»;
                      std::cout << <--output fichier» << std::setw(14) << « «
                              << «Spécifie le nom du fichier texte qui
contiendra l'Ascii Art.\n»;
```

```
std::cout << std::setw(30) << « «
                              << «Si ce paramètre n'est pas spécifié, l'Ascii
Art est sortie dans la console.\n»;
                      std::cout << «--palette fichier» << std::setw(13) << « «</pre>
                              << «Spécifie un fichier texte contenant la palette
de couleur Ascii.\n»;
                      std::cout << std::setw(30) << « «
                             << «Chaque ligne du fichier contient un
charactère en UTF - 8, du plus sombre au plus clair.\n»;
                      std::cout << std::setw(30) << « «
                             << «Si ce paramètre n'est pas spécifié, la
palette par défaut est \»W\», \»\», \»i\», \»:\», \»,\»,\»,\»,\»,\»,\»,\»
\n»;
                      std::cout << <--help> << std::setw(24) << « «
                              << «Affiche cette aide.»;
                      return 0;
              }
       for (int o = 0; o < argc; o++)
               std::cout << o << « : « << argv[o] << '\n';
              if (o == 1)
                      fichier = argv[o];
               if (o == 2)
                      fichier texte = argv[o];
              if (o == 3)
                      nom palette = argv[o];
              else
                      nom palette = «palette.txt»;
               if (o == 4)
                      largeur max = argv[o];
              else
                      largeur max = «1»;
              if (0 == 5)
                      hauteur max = argv[o];
              else
                      hauteur max = <1»;
    int hmax = std::stod(hauteur max);
    int lmax = std::stod(largeur max);
    std::ifstream image(fichier, std::ios base::binary);
    conversion (fichier, fichier texte, palette(nom palette),hmax, lmax);
```

Le programme version 5 compile, le fichier txt est créé mais rien n'est écrit dessus. Nous n'avons malheureusement pas eu le temps de le terminer. Vous trouverez donc sur Visual Studio la version 4 fonctionnelle.