GABRIEL Philippe

Groupe 503

Projet final du cours – Mario Bros.

Travail présenté à M. Desjardins

Informatique (ISI534-03)

Collège Beaubois

Jeudi 18 février 2016

Table des Matières

**Description générale du projet3**

Brève description3

Terminer un niveau4

Les ennemis4

Mort du joueur6

Les ennemis spéciaux6

Bouton « POW »6

Les points7

**Analyse du projet8**

L’interface8

Images8

Mouvement de Mario10

Le clavier et le mouvement dans le *Timer*12

Particularité dans le mouvement des ennemis12

Score record13

Option multijoueur13

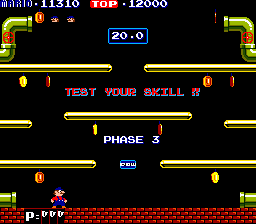
Musiques et effets sonores14

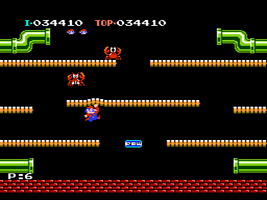
**Échéancier des différentes phases de développement14**

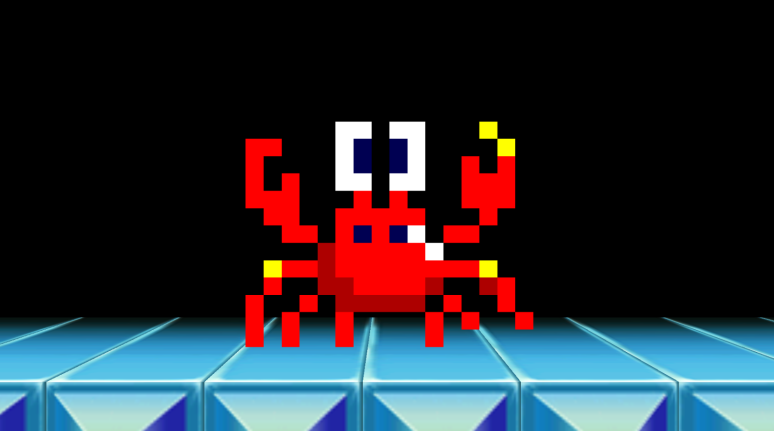
**Sources et références16**

**Description générale du projet**

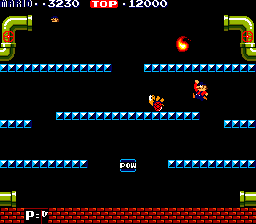
Le projet s’intitule Mario Bros. Le but du jeu est simple : survivre. Des créatures doivent être éliminés par le joueur niveau après niveau pour accumuler le plus de points possibles. Le joueur commence avec 2 vies.

Il existe des niveaux bonus dans lesquels il faut récupérer 10 pièces d’argent avant la fin du temps limite (20 secondes) pour recevoir une récompense en termes de points et une vie extra. Il est possible de jouer à deux dans le jeu ce qui peut alléger la difficulté. Les niveaux sont de petite taille, ils se déroulent sur le même écran. Les ennemis sont les seuls à pouvoir emprunter les tuyaux dans le niveau. Il existe deux tuyaux sur l’étage le plus haut, un à gauche l’autre à droite, d’où les ennemis font leur apparition. Lorsqu’ils arrivent à l’étage le plus bas, deux autres tuyaux les attendent, à gauche et à droite, pour les ramener à l’étage le plus haut. Le joueur ne peut utiliser ses tuyaux. Les seules mécanismes du joueur sont sauter et courir. Il est possible de sortir d’un côté de l’écran pour revenir de l’autre côté.

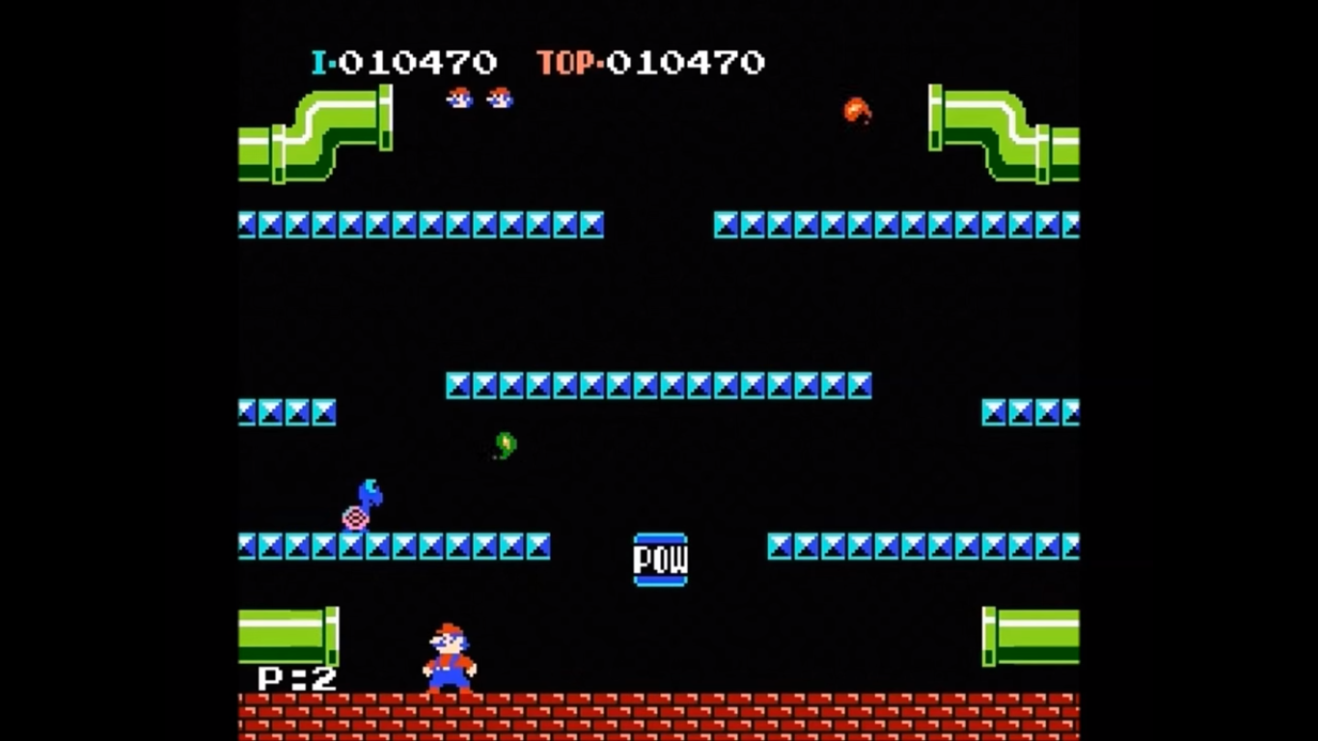


Pour terminer un niveau et passer à un autre, le joueur doit vaincre tous les ennemis à quelques exceptions. Des ennemis tels que des crabes, des tortues et des mouches se retrouvent fréquemment dans les niveaux et ce sont eux qui doivent se faire éliminer pour avancer dans les niveaux suivants. Pour ce faire, il doit frapper la plateforme à un moment où il se trouve directement en-dessous de l’ennemi en sautant de l’étage inférieur ce qui les retournera sur leurs dos (les crabes requièrent deux coups). Les mouches sont des ennemies pouvant faire de petits bonds pour avancer contrairement aux autres ennemis qui marchent sur sol. Il faut frapper la plateforme à un moment où les mouches sont sur le sol et directement en-dessous du joueur.





L’ennemi est alors dans sa position vulnérable et le joueur est alors en mesure de sauter sur la plateforme où il s’y trouve. Il peut alors le toucher pour l’éliminer du niveau. Si le joueur n’élimine pas l’ennemi assez rapidement, celui-ci se relèvera et adoptera une vitesse supérieure à son ancien état (il peut augmenter sa vitesse pour un total de deux fois). L’ennemi aura une couleur différente pour indiquer au joueur de son stade de vitesse. À noter que le dernier ennemi vivant du niveau adoptera automatiquement la vitesse maximale.



Le joueur meurt s’il touche directement n’importe quel ennemi. Il perd alors une vie et revient au niveau à l’aide d’une plateforme qui le ramène dans le plus haut étage du niveau. Il est possible pour le joueur de gagner une vie s’il est en mesure de ramasser toutes les pièces d’argent d’un niveau bonus (expliqué plus haut dans le document) qui lui donne ce bonus. Si le joueur perd toutes ses vies c’est la fin du jeu et son score se fait comparé au meilleur score achevé du jeu. Le meilleur des deux est choisi pour être affiché en haut au centre de l’écran.

Les niveaux ne comptent pas seulement comme ennemis des crabes, des tortues et des mouches. Quelques autres ne nécessitent pas de s’éliminer pour vaincre un niveau, mais ils accordent tout de même des points suite à leur expulsion du niveau. On retrouve par exemple des monstres de glace, des boules de feu vertes et rouge.

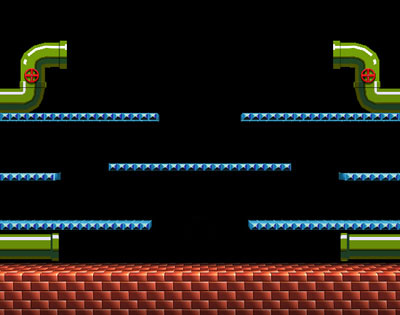
Ces ennemis reviennent dans un niveau d’une quantité indéfinie. Ce n’est qu’à la fin d’un niveau qu’ils disparaissent. Ils constituent surtout une distraction et augmentent le niveau de difficulté considérablement. Une arme spéciale retrouvée dans le jeu est le bouton « POW » qui élimine ou renverse selon le cas d’ennemi tout ennemi se retrouvant sur le sol d’un étage. Il peut être utilisé pour une limite de 3 fois et reviens chargé pleinement après un niveau bonus réussi ou non.

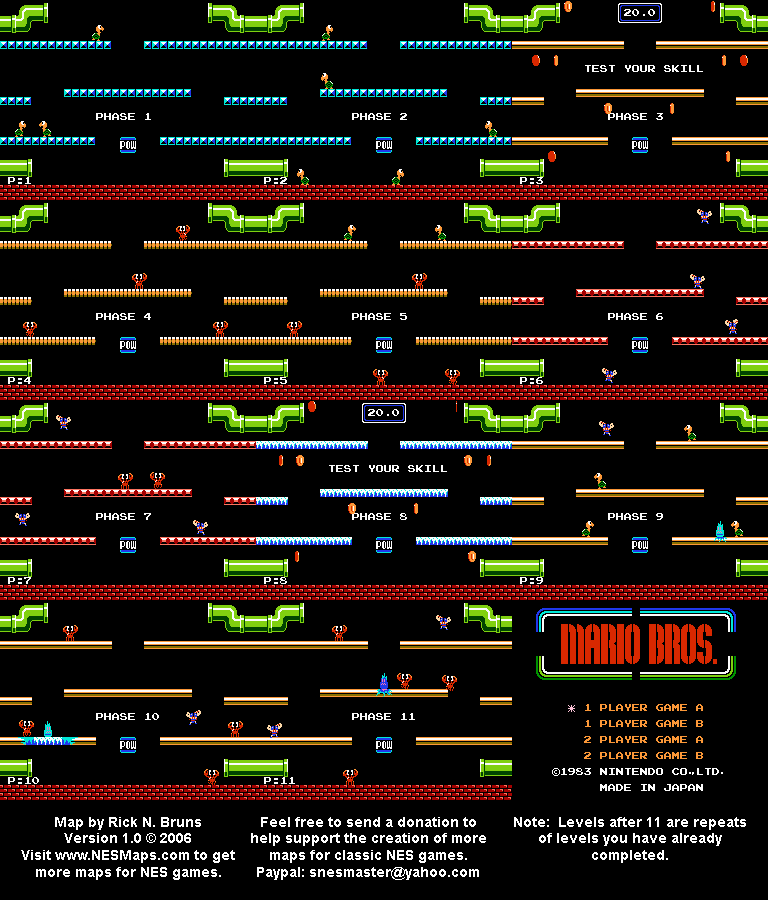
Voici un tableau démontrant les différentes quantités de points que le joueur peut acquérir ainsi que les manières de les acquérir :

|  |  |
| --- | --- |
| Variété de façons | Quantité de points |
| Renverser une tortue | 10 |
| Renverser un crabe | 10 |
| Renverser une mouche | 10 |
| Éliminer une tortue | 800 |
| Éliminer un crabe | 800 |
| Éliminer une mouche | 800 |
| Éliminer au moins deux ennemis de type tortue, crabe ou mouche en rapide successsion | Cette quantité va dans l’ordre d’une boucle For…Next où « n » représente le nombre d’ennemies et « p » le total de points amassés :  For i = 1 To n  p = p + 800\*i  Next i |
| Éliminer une boule de feu verte en utilisant la même stratégie que pour renverser des ennemis | 200 |
| Éliminer une boule de feu rouge en utilisant la même stratégie que pour renverser des ennemis | 1000 |
| Éliminer des montres de glace en utilisant la même stratégie que pour renverser les ennemis | 500 |
| Ramasser des pièces d’argent soit en allant l’amasser ou en frappant par en-dessous | 800 |
| Terminer un niveau bonus avec toutes les pièces d’argent | 5000 + 3000\*n (où « n » représente le nombre de niveaux bonus complétés sans avoir besoin d’être réussi avant le présent niveau) |

**Analyse du projet**

Ce projet contient de nombreuses parties à sa réalisation. Il s’avère être très long, mais avec une bonne gestion de temps, il ne devrait avoir aucun problème à le réaliser. Tout d’abord dans l’interface, on retrouve un canevas contenant le personnage principal du jeu : Mario. À l’aide du gestionnaire d’événements « Paint » de la fenêtre, on dessine l’arrière-plan du niveau. L’avantage de « Paint », contrairement à « Backdrop », c’est qu’il dimensionne automatiquement les images à la grandeur de l’objet où l’on veut dessiner. C’est cet arrière-plan que l’on utilisera :



Plusieurs images ont été modifiées à l’aide du logiciel Photoshop CS6 pour bien s’en servir dans le projet. On retrouve aussi quatre rectangles se situant au bout des quatre tubes retrouvés dans l’arrière-plan pour s’en servir comme repère lorsque des ennemis en sortiront ou en rentreront par exemple. Un autre rectangle est situé au même emplacement que le sol brun de l’arrière-plan pour s’en servir de repère lorsque le joueur ou des ennemis marcheront dessus. De plus, on retrouvera un canevas contenant le bouton « POW » au centre vers le bas de la fenêtre. Finalement des canevas situés aux emplacements des sols bleus pour s’en servir aussi de repères. La raison pourquoi ce sont des canevas et non des rectangles est que, à l’aide du gestionnaire d’événement « Paint », on retrouvera différents types de sol plus on avance dans les niveaux. Cette image nous démontre les différents niveaux possibles :

Il se retrouvera aussi des libellés pour contenir du texte principalement apportant quelques informations au joueur. La plupart de ces objets sont positionnés grâce au gestionnaire d’événement « Open » de la fenêtre. Voilà ce qui conclut les différents éléments de la bibliothèque d’objets dans l’interface du projet.

Dans ce projet, le joueur peut contrôler Mario avec les touches directionnelles du clavier pour uniquement courir à gauche et à droite ainsi que sauter. Seules les touches gauche, droite et haut seront donc utilisées. À l’aide d’un *Timer*. Mario bougera selon les actions que lui donne le joueur et en même temps, l’image du canevas où se trouve Mario changera selon la position où il se trouve lorsque le joueur lui applique des changements avec les touches directionnelles. Voici une image qui sera modifiée sur le logiciel PhotoShop CS6 pour convenir à nos besoins de la réalisation du projet :



Seules les images de Mario, du bouton « POW », des pièces d’argent et des deux boules de feux seront extraites par PhotoShop CS6 pour leur utilisation dans le projet. L’autre image nécessaire à l’accomplissement du projet est la suivante :



Les ennemis seront extrais de cette image ainsi que les éléments du menu du jeu, les différents types de sol, etc… à l’aide de PhotoShop CS6.

La manière dont va procéder le changement d’images est simple. Prenons comme exemple, Mario qui court vers la droite. Le joueur devrait peser sur la touche de droite pour accomplir cette action. Il faudrait tout d’abord créer une propriété du nom de « bouger » par exemple de type « Byte ». Dans le *Timer* en charge du mouvement, on écrit la condition *IF* qui requiert que le joueur pèse sur le bouton de droite. Selon le site d’aide de Xojo, cela s’avère être : « 124 ». Donc, la condition *IF* aura une allure de la sorte : « IF Keyboard.AsyncKeydown(124) THEN… ». Les exécutions qui suivent cette condition sont l’incrémentation de la propriété « bouger » de 1. À l’aide d’un *SELECT CASE* qui prend en considération la valeur de bouger, on change les images de canevas de Mario. Lorsqu’on arrive à la dernière image où Mario court, mais que le joueur pèse toujours sur la touche de droite, il faut ramener la valeur de « bouger » à 1, par exemple, ou tout autre valeur qui donnera l’impression que Mario court d’une manière continue. Si cependant, le joueur décide de relâcher la touche, il faudrait, logiquement que Mario arrête de courir et que l’image qui le montre debout apparaisse. C’est là que vient l’utilité du gestionnaire d’événement « KeyUp » de la fenêtre. Si le joueur aurait relâché la touche droite, la façon d’exprimer cette condition dans « KeyUp » est la suivante : « IF ASC(Key)=29 THEN… », pour ensuite rechanger l’image du canevas de Mario à celui où il se trouvait debout. C’est ce même raisonnement qui est utilisé pour la presque totalité des autres actions de Mario à quelques petites exceptions nécessaires de s’en rendre compte pour bien exécuter le projet.

Les ennemis de ce projet se retrouveront dans des canevas sur l’interface. Mais puisque le nombre d’ennemis est d’une quantité indéterminée et varie par niveaux, il serait mieux de créer des canevas à partir du code. Cette partie n’a pas tout à fait été découverte par moi, mais c’est de cette manière qu’il faudrait s’y prendre pour optimiser du mieux possible la gestion des éléments dans le projet. Le mouvement des ennemis varient selon l’ennemi en question et son stade de vitesse. Ce serait sans doute dans un *Timer* que le mouvement des ennemis sera traité. Quelques éléments importants à noter serait surtout lorsque les ennemis se font frapper par Mario par en-dessous. Il y a deux possibilités qui s’offrent. Premièrement, il est possible que le coup soit directement en-dessous de l’ennemi. Alors l’ennemi revolera dans les airs verticalement et se renversera sur lui-même. L’autre possibilité qui s’ajoute est si le coup n’est pas directement sous l’ennemi, qu’il soit fait un peu vers la droite ou la gauche. L’ennemi subira alors une trajectoire parabolique puisqu’en plus de sa vitesse à lui qui constituerait le vecteur horizontal, le coup lui applique un vecteur vertical. Selon la loi de Pythagore, la racine de la somme des carrés des vecteurs donnera le vecteur résultant. L’approche des vecteurs peut être utilisée pour former la trajectoire des ennemis ou alors l’approche d’une règle de degré deux qui décrira le mouvement. Une fonction pourrait effectuer ces calculs pour en ressortir la valeur désirée.

Le joueur a une possibilité d’achever un score record qui serait affiché au centre en haut de la fenêtre et qui reste sauvegardé dans un fichier texte. À chaque fois que le joueur perd, son score est comparé à celui affiché dans le record. Le plus grand est celui qui s’écrit dans le fichier texte et qui s’affiche dans le jeu. Évidemment, lors du premier essai du jeu, le score record est de zéro, donc suite à la comparaison des scores, c’est le score du joueur qui remplacera ce zéro. Une méthode « sub » sera utilisé pour prendre en paramètre le score final du joueur et gérer le fichier texte pour en ressortir la valeur record pour ensuite comparer. Les notes de cours sur les fichiers ainsi que des programmes exemples leur ayant recours sont une excellente source pour se remémorer de ces vieilles notions.

Autre fonctionnalité qui serait intéressante d’ajouter, mais qui ne constitue pas une priorité, est celle d’un deuxième joueur, Luigi, pour assister Mario de façon à alléger la difficulté du jeu. Lui aussi aura un score ainsi que des vies propres à lui. Les deux joueurs peuvent jouer soit pour s’entraider ou pour compétitionner.

Dernière particularité du projet est l’ajout de musique. Par exemple, il pourrait y avoir de la musique dans le menu et une différente lorsque le joueur débute un niveau, etc… Les effets sonores sont très intéressants aussi. Il pourrait y en avoir lorsque le joueur élimine un ennemi ou lorsqu’il marche tout simplement, etc…

**Échéancier des différentes phases de développement**

|  |  |
| --- | --- |
| Date de cours | Phases de développement |
| 26 Janvier – Jour 7 | Interface : Positionnement des rectangles et des canevas.  Intègrement de quelques images modifiées sur PhotoShop CS6. |
| 1 Février – Jour 2 | Mouvement de Mario : Il peut marcher vers la droite.  Intègrement de quelques images modifiées sur PhotoShop CS6. |
| 2 Février – Jour 3 | Mouvement de Mario : Il peut marcher parfaitement de gauche à droite.  Intègrement de quelques images modifiées sur PhotoShop CS6. |
| 3 Février – Jour 4 | Mouvement de Mario : Il peut sauter verticalement seulement.  Intègrement de quelques images modifiées sur Photoshop CS6. |
| 10 Février – Jour 7 | Mouvement de Mario : Il peut sauter verticalement et vers la gauche et la droite. |
| 16 Février – Jour 2 | Mouvement de Mario : Ajustement de quelques erreurs avec le saut. |
| 17 Février – Jour 3 | Mouvement de Mario : Bouge et saute parfaitement, quelques ajustements peuvent être ajoutés plus tard.  Interface : Quelques éléments ont eu leur position modifiée. |
| 18 Février – Jour 4 |  |
| 23 Février – Jour 7 | Collisions avec les étages et déplacements dans le niveau soient complétés |
| 7 Mars – Jour 2 |  |
| 8 Mars – Jour 3 | Configuration du bouton « POW » avec ses effets |
| 9 Mars – Jour 4 |  |
| 14 Mars – Jour 7 |  |
| 23 Mars – Jour 2 | Mouvements des ennemis crabes et tortues |
| 24 Mars – Jour 3 |  |
| 4 Avril – Jour 4 |  |
| 7 Avril – Jour 7 | Mouvement des mouches |
| 13 Avril – Jour 2 |  |
| 14 Avril – Jour 3 |  |
| 15 Avril – Jour 4 | Renversement des ennemis (propulsement dans les airs) |
| 20 Avril – Jour 7 |  |
| 27 Avril – Jour 2 | Mouvement des ennemies boulles de feu rouge et vert, monstre de glace |
| 28 Avril – Jour 3 |  |
| 29 Avril – Jour 4 |  |
| 4 Mai – Jour 7 | Régler les scores et faire la fenêtre du menu |
| 10 Mai – Jour 2 | Introduire tous les niveaux et leurs différences |
| 11 Mai – Jour 3 | Enregistrement du score record |
| 12 Mai – Jour 4 |  |
| 17 Mai – Jour 7 | Musique dans le jeu |
| 25 Mai – Jour 2 |  |
| 26 Mai – Jour 3 |  |
| 27 Mai – Jour 4 |  |
| 1 Juin – Jour 7 | Ajout de la fonction multijoueur |
| 7 Juin – Jour 2 |  |

# Sources et références

* <http://2.bp.blogspot.com/-WBYn_fgeSsc/VLPkmffcH1I/AAAAAAAAyW8/lVbkNjtERCo/s1600/Mario_Bros_(Arcade)_10.gif>
* <http://www.nindb.net/nes/mario-bros-classic-series/img/screenshot.png>
* <http://www.mariowiki.com/images/3/38/Shellcreeper-SSBBrawl.png>
* <http://www.mariowiki.com/images/a/ad/Sidestepper-SSBBrawl.png>
* <http://playinstantgame.com/wp-content/plugins/RSSPoster_PRO/cache/392fc_Fighter-fly.png>
* <https://upload.wikimedia.org/wikipedia/en/c/cf/Mario_Bros._Gameplay.gif>
* <https://www.youtube.com/watch?v=JHlUEJXaxfY>
* <https://www.youtube.com/watch?v=fKqC8tPp-H0>
* <http://cdn.wikimg.net/strategywiki/images/5/51/MB_Icicle.png>
* <http://cdn.wikimg.net/strategywiki/images/1/18/Mario_Bros_NES_slipice_new.png>
* <http://www.mariowiki.com/images/2/22/MB_fireballs.jpg>
* <http://vignette2.wikia.nocookie.net/ssb/images/6/6c/MarioBros_Stage.jpg/revision/latest?cb=20080212071306>
* <http://www.vgmaps.com/Atlas/NES/MarioBros-Maps.png>
* <http://www.mariomayhem.com/downloads/sprites/mario_bros_nes/mario_bros_nes_sprite_sheet_2.png>
* <http://www.mariomayhem.com/downloads/sprites/mario_bros_nes/mario_bros_nes_sprite_sheet.png>