New Scripts

Many scripts from TDD work in MFP, but some of them have been removed (e.g. SetInventoryDisabled).
However, MFP has a set of its own new scripts:

Player

```
void SetLanternFlickerActive(bool abActive);
```

Enables the lantern flicker effect.

abActive - set to true to enable the effect

```
void SetPlayerInfection(float afAmount);
void AddPlayerInfection(float afAmount);
float GetPlayerInfection();
```

Infection related scripts.

Screen effects

```
void ShowScreenImage(string asImage, int alPosX, int alPosY, float
afUnknown, bool abUnknown2, float afTime, float afFadeIn, float afFadeOut);
```

Displays an image on the screen. Originally used to show the MFP logo in-game.

asImage - the image to display. E.g. `startup_aamfp_logo.jpg`
alPosX - horizontal position of the image. 0 is right screen border, smaller values are left. -400 was used for the middle of the screen.
alPosY - vertical position of the image. 0 is bottom, smaller values are up. -350 was used for the middle of the screen.
afUnknown - Unless set to below 0, the image won't appear. Might have more functionality to it.
abUnknown2 - setting this to true makes the image not appear. Might have more functionality to it.
afTime - image display time.
afFadeIn - fade in time. It's added to the base display time.
afFadeOut - fade out time. It's added to the base display time.

Enemies

```
void SetEnemyMoveType(string &in asEnemy, string asMoveType);
```

❓**This script hasn't been tested.**

Sets the enemy move type.

asEnemy - the in-game enemy entity

asMoveType - options include: WalkBiped, RunBiped, ChargeBiped, and probably Quadruped variations as well.

```
void SetManPigType(string &in asEntity, string &in asType);
```

This script hasn't been tested.

Seems to alter the enemy AI. It was mostly used in conjunction with Child enemies (but also with Wretches). Presumably it changes the AI behaviour to run from the player.

asEntity - the enemy in question. Can be Enemy_ManPig or Enemy_Child type.

asType - only “Freddy” has been used in the entire game. It is unknown what other options are.

Other

```
void SetParticleSystemActive(string &in asParticle, bool abX);
```

This script hasn't been tested.

```
void AddHint(string &in asEntity, string asUnknown);
```

This script hasn't been tested.

Might be a replacement for GiveHint from TDD.
Another option is that it gives a specified hint (*asUnknown*) upon touching the *asEntity*.

```
SetLightVisible(string& asLightName, bool abVisible);
```

This script hasn't been tested.

Enables/disables lights.

It was used in the main game scripts (in conjunction with SpotLights), which might indicate that unlike in TDD, this time it actually works.

asLightName - internal name
abVisible - determines the state of the light

```
void SetPhysicsAutoDisable(string &in asEntity, bool abX);
```

This script hasn't been tested.

Unknown usage.

asEntity - entity in question. It was used with chandelier_nice.

abX - whether to disable the object physics (*when?*).

From:
https://wiki.frictionalgames.com/ - **Frictional Game Wiki**

Permanent link:
https://wiki.frictionalgames.com/hpl2/machine_for_pigs/new_scripts?rev=1583596910

Last update: **2020/03/07 16:01**