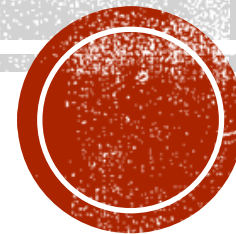


# ASSISANT DE BUREAU

Projet de Darren Lambert et Léo Mercier



# SETUP

Notre programme doit se lancer au démarrage de Windows

```
Write-Output "Execution de l'automatisation du BOT ...."

#####Initialisation#####

$TaskName = "BotAutomatisation"

#####Fonction#####

function WaitForState
{
    param( $State )

    $Info = Get-ScheduledTask -TaskName $TaskName
    while( $State -ne $Info.State)
    {
        Start-Sleep -Seconds 1
        $Info = Get-ScheduledTask -TaskName $TaskName
    }
}

#####Execution#####

$Now = Get-Date
$Time = $Now.AddMinutes(1)

$action = New-ScheduledTaskAction -Execute "PowerShell.exe" -Argument ( "C:\Users\533\Documents\GitHub\AssitantPowershell\MainScriptAssistant.ps1" )
$Desc = " Bot Autonome "
#$Trigger = New-ScheduledTaskTrigger -Once -At $Time
$Trigger = New-ScheduledTaskTrigger -AtLogon

$Task = New-ScheduledTask -Action $Action -Description $Desc -Trigger $Trigger

Register-ScheduledTask -InputObject $Task -TaskName $TaskName
if( -Not $? )
{
    Write-Error " La Configuration à échoué"
    exit 10
}

WaitForState -State Running

Write-Output " La tâche est en cours d'execution "

WaitForState -State Ready

Write-Output " La tâche s'est terminé "

Unregister-ScheduledTask -TaskName $TaskName
if( -Not $? )
{
    Write-Error " La Configuration à échoué"
    exit 10
}

Write-Output OK
```

Création de la tâche programmé

Fonction permettant de suivre l'exécution de la tâche programmé

Suivi de l'exécution de la tâche



# PROFIL ÉTUDE

```
$Desktop = $env:USERPROFILE + "\Desktop\  
$site = ('https://cegepba.omnivox.ca/Login/Account/Login', 'https://medium.com/tag/programming', 'https://github.com/login')  
$notesFile = $Desktop + "NOTES_$((Get-Date).ToString('yyyy-MM-dd')).txt"  
Set-Location ($env:USERPROFILE + "\Documents\GitHub\AssitantPowershell\")
```

Ouverture des sites web de la liste

```
try  
{  
    ForEach($s in $site)  
    {  
        Start-Process $s  
        Start-Sleep -Seconds 1  
    }  
}  
catch  
{  
    $wshell = New-Object -ComObject Wscript.Shell  
    $Output = $wshell.Popup("An error occur while opening websites")  
}
```

Ouverture de VisualStudio code et d'une note  
notepad++ nommé avec la date d'aujourd'hui et  
enregistrer sur le bureau

```
try  
{  
    # open visual studio code  
    Start-Process -WindowStyle Hidden code .  
    # open notepad++ with textfile named by date of the day and save on the desktop  
    Start-Sleep -Seconds 2  
    start notepad++ $notesFile  
}  
catch  
{  
    $wshell = New-Object -ComObject Wscript.Shell  
    $Output = $wshell.Popup("An error occur while opening VScode and notepad++")  
}
```

Minimisation de toutes les fenêtres

```
Start-Sleep -Seconds 2  
$shell = New-Object -ComObject "Shell.Application"  
$shell.minimizeall()  
Start-Sleep -Seconds 2
```

Lancement du script  
SetWallPaper.ps1 pour changer le  
fond d'écran.

```
try  
{  
    .\SetWallPaper.ps1 "C:\Users\533\Documents\GitHub\AssitantPowershell\img.jpg" "Stretch"  
}  
catch  
{  
    $wshell = New-Object -ComObject Wscript.Shell  
    $Output = $wshell.Popup("An error occur while setting wallpaper")  
}
```



# INTERFACE DU BOT

```
try
{
    #====Section Création Forms=====
    Add-Type -AssemblyName System.Windows.Forms
    [System.Windows.Forms.Application]::EnableVisualStyles()
    $Form = New-Object System.Windows.Forms.Form
    $Form.ClientSize = '300,300'
    $Form.Text = "MonteBot Interface"

    #====Section Education=====
    $BoutonEducation = New-Object System.Windows.Forms.Button
    $BoutonEducation.Location = New-Object System.Drawing.Point(100,100)
    $BoutonEducation.Width = 80
    $BoutonEducation.Height = 40
    $BoutonEducation.Text = "Session Etude"

    #====Section Gaming=====
    $BoutonGaming = New-Object System.Windows.Forms.Button
    $BoutonGaming.Location = New-Object System.Drawing.Point(100,150)
    $BoutonGaming.Width = 80
    $BoutonGaming.Height = 40
    $BoutonGaming.Text = "Session Gaming"

    #====Section Fermer Session=====
    $BoutonClose = New-Object System.Windows.Forms.Button
    $BoutonClose.Location = New-Object System.Drawing.Point(100,200)
    $BoutonClose.Width = 80
    $BoutonClose.Height = 40
    $BoutonClose.Text = "Débrancher MonteBot"

    #====Section Label=====
    $Label = New-Object System.Windows.Forms.Label
    $Label.Location = New-Object System.Drawing.Point(100,50)
    $Label.Size = '100,100'
    $Label.Text = "bonjour"

    $Title = New-Object System.Windows.Forms.Label
    $Title.Font = [System.Drawing.Font]::new("Microsoft Sans Serif", 12, [System.Drawing.FontStyle]::Bold)
    $Title.Location = New-Object System.Drawing.Point(100,30)
    $Title.Text = "MonteBot"
}
catch
{
    $wshell = New-Object -ComObject Wscript.Shell
    $Output = $wshell.Popup("Erreur: Un Problème est apparu lors de la création d'un ou plusieurs Composant du Forms")
}
```

Permet de personnaliser la  
taille, le positionnement, etc...



# EVENEMENT CLICK

```
#====Section Event Click=====
```

```
$BoutonEducation.Add_Click({  
    $Label.Text = "Passons au choses sérieuse !!!"  
    try  
    {  
        MonteBot -sayTheLine "Passons au choses sérieuse !!!" -rate 0  
  
        Set-Location "C:\Users\533\Documents\GitHub\AssitantPowershell\  
        .\ScriptProfilEtude.ps1  
    }  
    catch  
    {  
        $wshell = New-Object -ComObject Wscript.Shell  
        $Output = $wshell.Popup("Erreur: Initialisation de la session Education impossible")  
    }  
})
```

```
$BoutonGaming.Add_Click({  
    $Label.Text = "Gaming Lets go"  
  
    try  
    {  
        MonteBot -sayTheLine "Gaming Lets go" -rate 5  
  
        Set-Location "C:\Users\533\Documents\GitHub\AssitantPowershell\  
        .\ScriptProfilGaming.ps1  
    }  
    catch  
    {  
        $wshell = New-Object -ComObject Wscript.Shell  
        $Output = $wshell.Popup("Erreur: Initialisation de la session Gaming impossible")  
    }  
})
```

```
$BoutonClose.Add_Click({  
    $Label.Text = "-_- Goodbye "  
    try  
    {  
        MonteBot -sayTheLine "Good bye " -rate -1  
        $Form.Close()  
    }  
    catch  
    {  
        $wshell = New-Object -ComObject Wscript.Shell  
        $Output = $wshell.Popup("Erreur: Impossible de quitter le programme, Veuillez Faire Alt + F4")  
    }  
})
```

Permet d'ajouter des  
événement de Click pour  
pouvoir lancé d'autres script.

Permet d'arreter le programme  
Powershell



# INITIALISATION DU FORMS

```
MonteBot -sayTheLine "Bonjour maître, Comment allez-vous ? !!!" -rate 0
```

```
try
{
    $Form.controls.AddRange(@($BoutonEducation,$Label,$BoutonGaming,$Title,$BoutonClose))
    $Form.ShowDialog()
}
catch
{
    $wshell = New-Object -ComObject Wscript.Shell
    $Output = $wshell.Popup("Erreur: Le Sytème n'arrive pas à afficher l'interface")
}
```

Cette ligne de code à pour but d'ajouter chaque composant crée dans le Form.

Permet d’Afficher le Forms

```
function MonteBot{
    param(
        $sayTheLine,
        $rate
    )

    try
    {
        Add-Type -AssemblyName System.speech
        $speak = New-Object System.Speech.Synthesis.SpeechSynthesizer
        $speak.SelectVoice('Microsoft Hortense Desktop')
        $speak.Rate = $rate
        $speak.Speak($sayTheLine)
    }
    catch
    {
        $wshell = New-Object -ComObject Wscript.Shell
        $Output = $wshell.Popup("Erreur: Le Bot n'arrive plus à Communiquer avec nous")
    }
}
```

Pour éviter la redondance j’ai crée une fonction pour les commandes vocale du Bot

Cette section nous permet de choisir la voix du Bot, ainsi que sa vitesse de parole.

