

## Sprite

### Opdrachtbeschrijving

De opdracht bestaat uit het samenvoegen van een sprite en een figuur.

### Stappenuitleg basisopdracht:

Eerst lezen we de originele afbeelding en de sprite in.

```
% Laad de originele image
orig_file = get(handles.editOrig, 'String');

if ~isempty(orig_file)
    im_orig=imread(char(orig_file));
    set(handles.imgOrig, 'HandleVisibility', 'ON');
    axes(handles.imgOrig);image(im_orig);
    axis equal;
    axis tight;
    axis off;
    set(handles.imgOrig, 'HandleVisibility', 'OFF');
end;

% Laad de sprite
sprite_file = get(handles.editSprite, 'String');

if ~isempty(sprite_file)
    %teken de sprite
    im_sprite=imread(char(sprite_file));
    set(handles.imgSprite, 'HandleVisibility', 'ON');
    axes(handles.imgSprite);image(im_sprite);
    axis equal;
    axis tight;
    axis off;
end;
```

Dan gaan we de mask zoeken van onze afbeelding, bijvoorbeeld Grote smurf. Dit doen we door de sprite te kopiëren:

```
%bereken de mask
im_mask = im_sprite;
maskdim = size(im_mask);
```

Daarna maken we de geselecteerde kleur zwart en de overige kleuren wit.

```
for i = 1:maskdim(1)
    for j = 1:maskdim(2)
        mask = (RGB(1,1) > im_mask(i,j,1)) + (RGB(1,2) <
im_mask(i,j,1)) + (RGB(2,1) > im_mask(i,j,2)) + (RGB(2,2) < im_mask(i,j,2))
+(RGB(3,1) > im_mask(i,j,3)) + (RGB(3,2) < im_mask(i,j,3));
        if mask > 0
            im_mask(i,j,:) = 254;
        else
```

```

        im_mask(i,j,:) = 0;
    end;
end;
end;

%teken mask
set(handles.imgMask, 'HandleVisibility', 'ON');
axes(handles.imgMask);image(im_mask);
axis equal;
axis tight;
axis off;
end;

```

Om het resultaat te berekenen. Lezen we de x en y coördinaten in. We gaan dan bij de mask kijken of het om een zwarte of een witte pixel gaat. Indien de pixel wit is gaan we de gewenste pixel van het origineel vervangen door de overeenkomende pixel van de sprite.

```

%teken het resultaat
set(handles.imgResult, 'HandleVisibility', 'ON');
axes(handles.imgResult);image(im_result);
axis equal;
axis tight;
axis off;

RGBset(1) = str2num(get(handles.editR, 'String')) / 254;
RGBset(2) = str2num(get(handles.editG, 'String')) / 254;
RGBset(3) = str2num(get(handles.editB, 'String')) / 254;

set(handles.editColor, 'BackgroundColor', RGBset);

% Zet alle velden terug uit.
set(handles.imgSprite, 'HandleVisibility', 'OFF');
set(handles.imgOrig, 'HandleVisibility', 'OFF');
set(handles.imgMask, 'HandleVisibility', 'OFF');
set(handles.imgResult, 'HandleVisibility', 'OFF');

```

Omdat we de beelden niet steeds willen opnieuw willen inlezen en de mask niet steeds willen berekenen. Gebruiken we 3 globale waarden voor de afbeeldingen. Dit plaatsen we bovenaan elke functie waar we de beelden nodig hebben.

```

%afbeeldingen globaal maken
global im_orig
global im_sprite
global im_mask

```

We laten de gekozen maskeer kleur zien in een tekstvak. Dit doen we als volgt:

```

%teken de kleur
RGBset(1) = str2num(get(handles.editR, 'String')) / 254;
RGBset(2) = str2num(get(handles.editG, 'String')) / 254;
RGBset(3) = str2num(get(handles.editB, 'String')) / 254;
set(handles.editColor, 'BackgroundColor', RGBset);

```

Deze functies komen meermaals voor in het programma. Elke knop bevat de functies die hij moet uitvoeren.

## Het resultaat

Hieronder een voorbeeld van het resultaat:

