Minimax, Alpha-Beta, Negamax

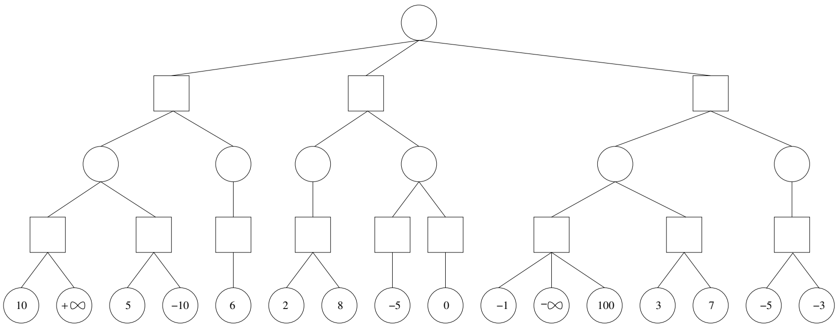
N. Durand, D. Gianazza

How to get started:

- Edit file mmtd.ml
- To compile the file: *ocamlc mmtd.ml* or *ocamlopt mmtd.ml*
- To execute the program: *./a.out*
- The objective is to write a minimax function, an alpha-beta function, and eventually a negamax function and to test it on the example seen in class.
- The functions are in the file and need to be completed. Each time a leaf is explored a counter should be incremented (count1 for minimax, count2 for alplhabeta and count3 for negamax).

Minimax(t)

```
match $t$ with
Leaf x:
    return x
Nodemin $l_{min}$:
    m:=$+\infty$
    for all $t_{min} \in l_{min}$ do
        m:=min m Minimax($t_{min}$)
    end for
    return m
Nodemax $l_{max}$:
    m:=$-\infty$
    for all $t_{max} \in l_{max}$ do
        m:=max m Minimax($t_{max}$)
    end for
    return m
end match
```



alphabeta(t, a, b)

```
match $t$ with
Leaf s:
    return s
Nodemin $l_{min}$:
    s:=$b$
    for all $t_{min} \in l_{min}$ do
        s:=min s alphabeta($t_{min}$, $a$, $s$)
        if $s \leq a$ then
            return s
        end if
    end for
    return s
Nodemax $l_{max}$:
    s:=$a$
    for all $t_{max} \in l_{max}$ do
        s:=max s alphabeta($t_{max}$, $s$, $b$)
        if $s \geq b$ then
            return s
        end if
    end for
    return s
end match
```

negamax(t, a, b)

```
match $t$ with
Leaf s:
    return s
Node $l$:
    s:=$a$
    for all $t \in l$ do
        s:=max s -negamax($t$, $-b$, $-s$)
        if $s \geq b$ then
            return s
        end if
    end for
    return s
end match
```

For those who are interested

Try to solve the tic-tac-toe problem using the previous functions. The tree can be created dynamically. The difficult part is to detect leaves in the tree and write the leaf evaluation.