

# Метод Монте-Карло пошуку у структурах даних гри Го

Виконав  
студент ФПМ гр. КП-4 Іс  
Крамаренко Олексій Андрійович

Керівник дипломного проекту  
к.т.н., доцент Дідковська М.В.

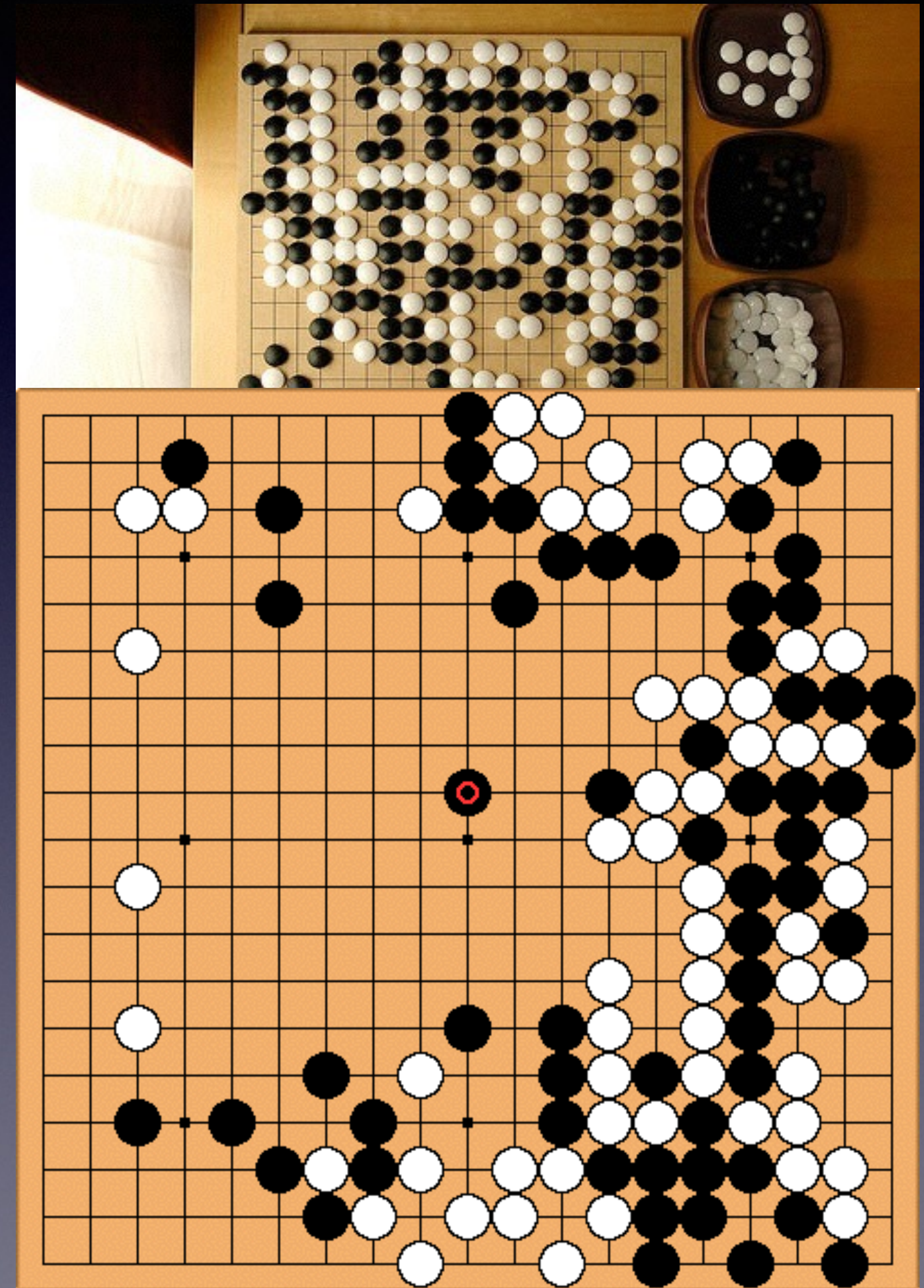


# Актуальність програмної розробки

- Більшість розробок – комерційні
- Некомерційні продукти – закриті
- Комп'ютери грають у Го погано

Основні причини:

- Розмір дошки
- При розвитку гри аналіз значно ускладнюється
- Складність створення точної функції оцінки





# Існуючі рішення та їх недоліки

- CrazyStone – комерційна
- The Many Faces of Go – комерційна
- MoGo – закрита
- GNU Go – неактуальна
- MasterGo – комерційна
- Kombilo – не модульна



# Мета проекту

Метою проекту є створення відкритої модульної системи для роботи зі структурами даних гри Го та надання користувачу можливостей для паралельного пошуку ходів та шаблонів у цих структурах.



# Основні задачі системи

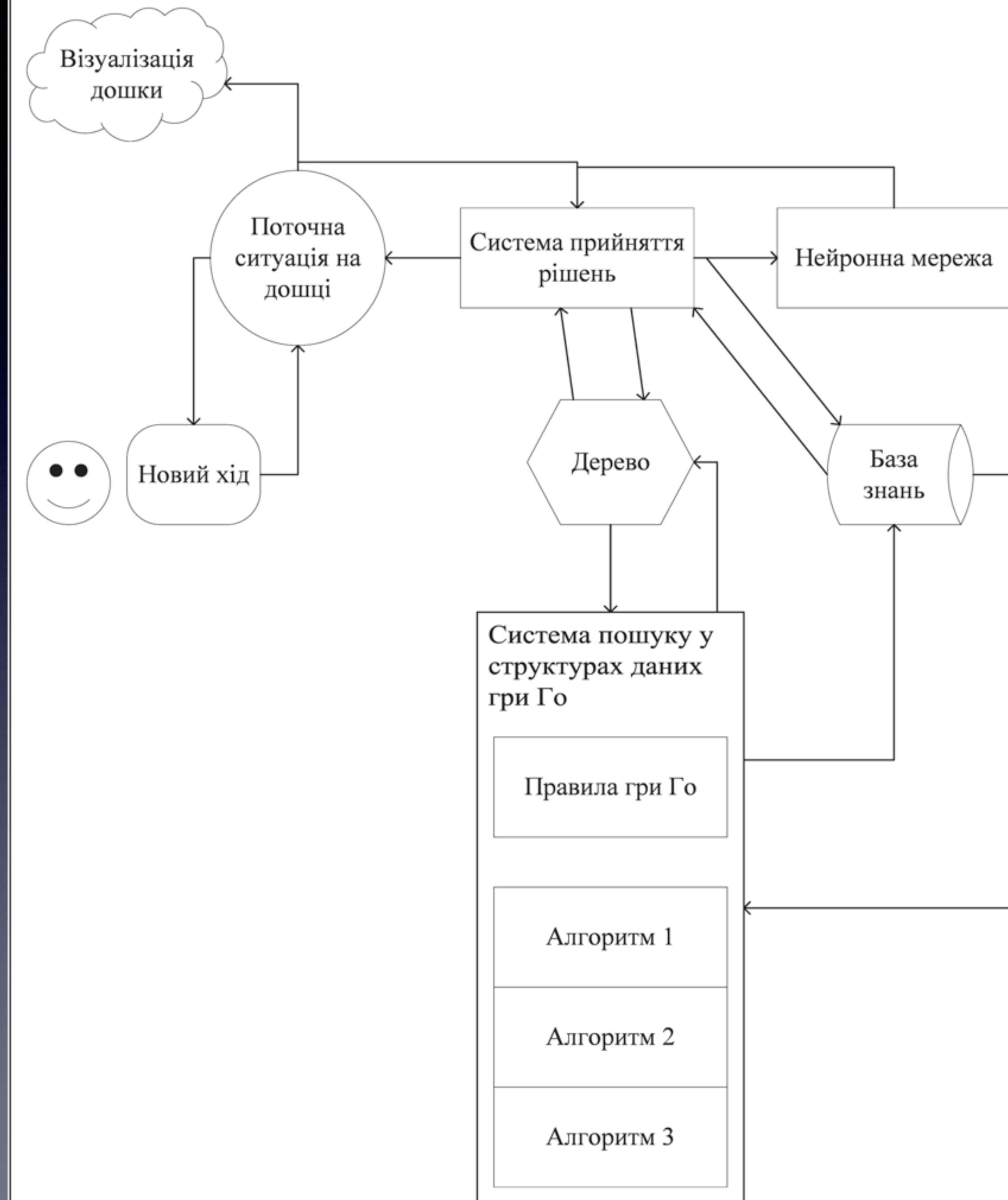
- Читання SGF-файлів, перетворення їх у внутрішнє дерево
- Пошук у дереві партії методами
  - Монте-Карло
  - Модифікацією Альфа-Бета відсічення
  - Пошуком за шаблоном



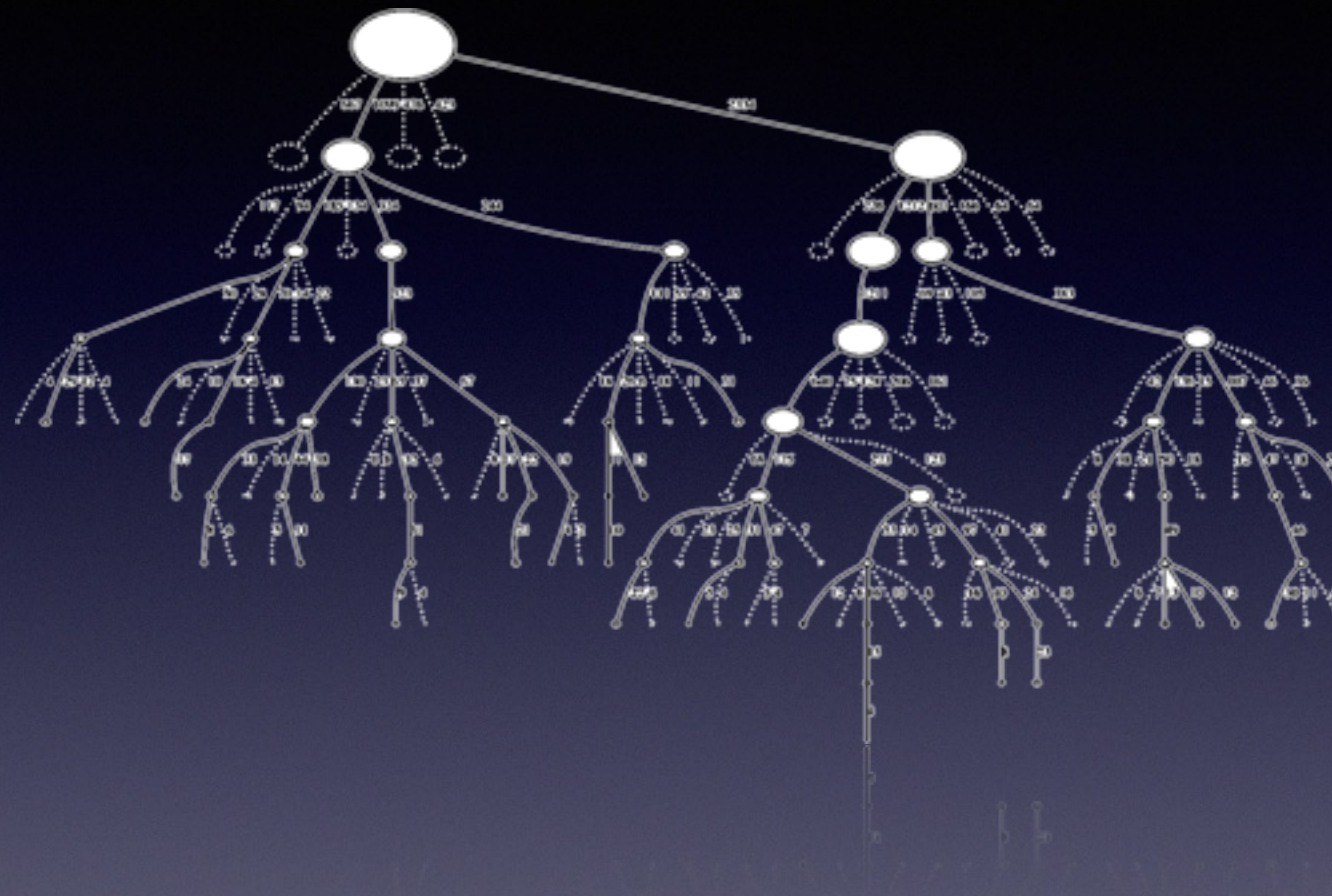
## Основні задачі системи:

- Реалізація взаємодії між модулями використовуючи дерево гри
- Генерація набіру можливих ходів використовуючи модуль політики
- Пошук оптимальних наступних ходів

Система гри в Го







# Метод Монте-Карло пошуку в дереві



# Метод верхньої оцінки значущості для дерев

Значущість вузла зменшується кожен раз, коли його відвідують

Значущість вузла збільшується кожен раз, коли відвідують інший дочірній вузол  
батьківського вузла

$$\text{UCTValue}(\text{parent}, n) = \text{winrate} + \sqrt{\ln(\text{parent.visits}) / (5 \times n.\text{nodevisits})}$$

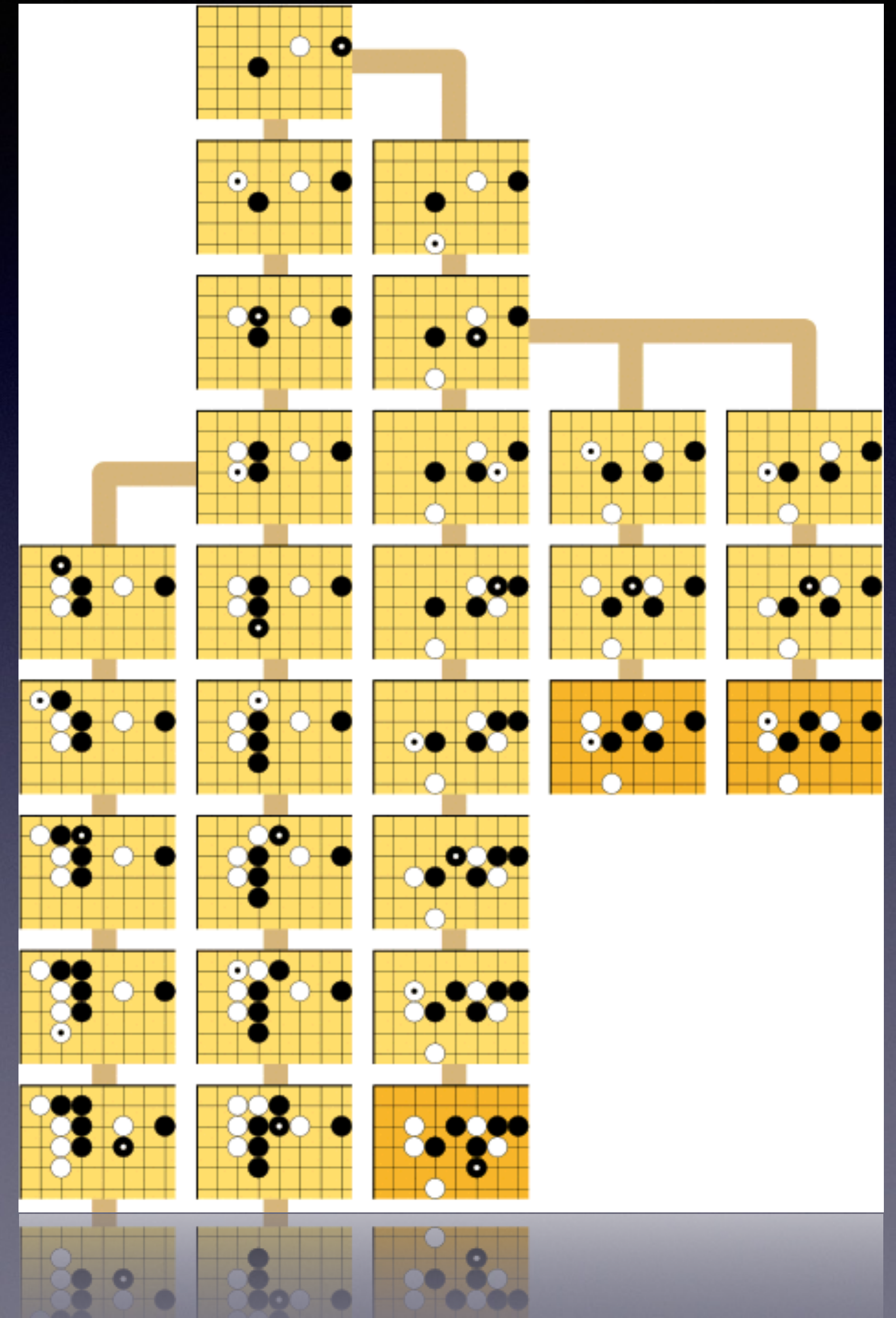


## Пошук у дереві використовуючи базу шаблонів

```
for each game record R in the game record collection do
  for each move M in R do
    Play M on the game board
    Obtain the 5-by-5 region R centered by M
    Rotate and flip R into its canonical form

    if R is not in our pattern database then
      Add R into the pattern database
      Set frequency number of R to be 0
    end if

    Increase the frequency number of R by 1
  end for
end for
```





# Альфа-бета відсічення

Input: Color: color, Integer: depth, Integer:  $\alpha$ , Integer:  $\beta$

Output: Integer: value of best play

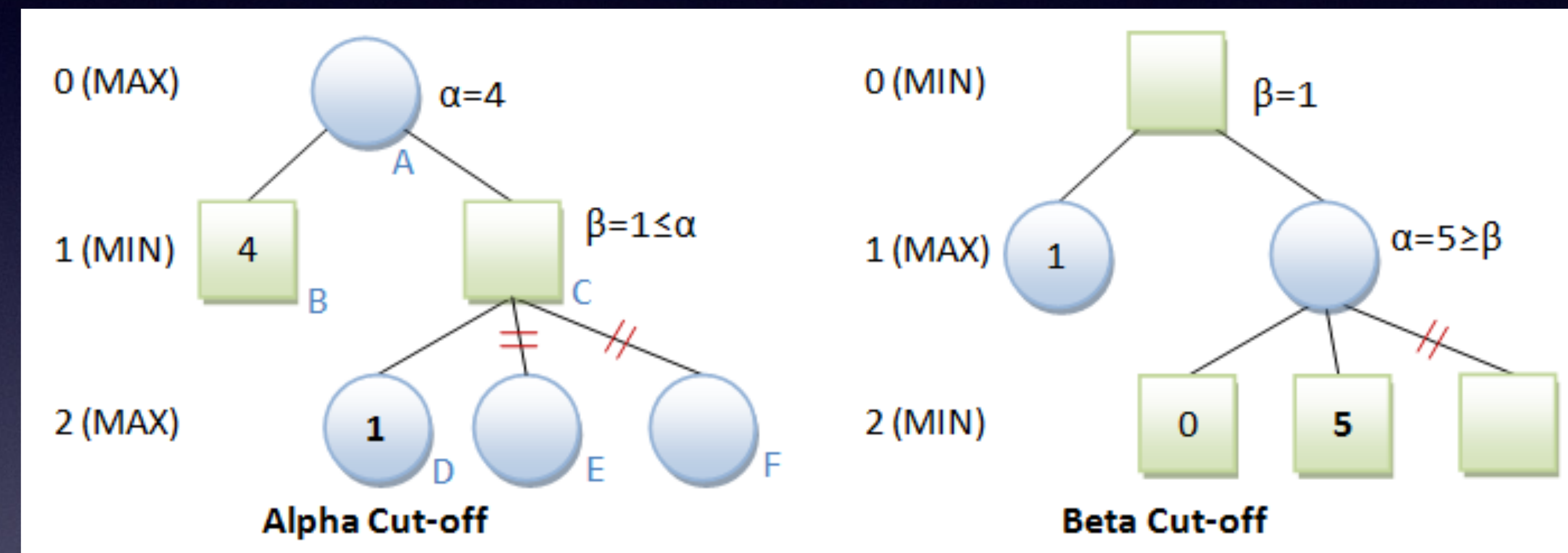
```

1:  if depth = 0 then
      return Evaluate(color)
    end if

    moves ← GenerateMoves()
5:  for every move in moves do
      makeMove(move)
      eval ← -AlphaBeta(-color, depth-1, - $\beta$ , - $\alpha$ )
      unmakeMove(move)

      if eval ≥  $\beta$  then
10:         return  $\beta$ 
      end if

      if eval >  $\alpha$  then
           $\alpha$  ← eval
          if depth = defaultDepth then
15:             bestmove ← move
          end if
      end if
    end for
    return  $\alpha$ 
  
```





# Порівняння з існуючими продуктами

- Проект відкритий, програмний код зберігається на Github
- Система працює на платформі Java
- Система модульна, кожен модуль можна використовувати окремо





# Приклад роботи системи

```
java
sgfc.core=> (first (ssimplify (simplify (sgf2tree "resources/kramar.sgf"))))
("pp" "dd" "fp" "cn" "dq" "cq" "dr" "cr" "ep" "do" "cf" "ch" "fc" "eb" "cc" "cd" "dc" "ec"
"ed" "fd" "ee" "gc" "bd" "be" "ce" "bc" "de" "db" "cb" "kc" "bg" "bh" "da" "fb" "qf" "nq"
"np" "mq" "oq" "or" "pr" "os" "ps" "mr" "hq" "jq" "lp" "lq" "mp" "kp" "ko" "jo" "kn" "jn"
"pl" "fn" "ip" "jp" "ir" "jr" "qc" "pc" "qd" "pe" "qe" "pf" "pg" "og" "ph" "oh" "oi" "ni"
"oj" "nj" "nk" "mk" "nl" "ml" "mm" "km" "lm" "ll" "kl" "kk" "jm" "jk" "mj" "lk" "mi" "nh"
"of" "nf" "mh" "oe" "ih" "kh" "kg" "lg" "jh" "ki" "lf" "mg" "ji" "kf" "jg" "le" "kj" "lh"
"jj" "lj" "ik" "li" "jl" "jf" "if" "fh" "fg" "gh" "gg" "eh" "fk" "gk" "gl" "gj" "fj" "fl"
"fm" "el" "em" "dm" "gm" "en" "gi" "hi" "fi" "hl" "hk" "hj" "hm" "ek" "hh" "ej" "id" "ic"
"jd" "jc" "kd" "ld" "ke" "mf" "je" "lf" "eg" "hd" "ie" "he" "dh" "ei" "di" "cj" "ci" "bi"
"dj" "ck" "cg" "dk" "dg" "io" "ho" "iq" "hr" "hp" "gp" "hn" "go" "gn" "im" "fr" "fs" "es"
"gs" "gr" "hs" "eq" "fq" "er" "pb" "ob" "qb" "oa" "pa" "pi" "pj" "qi" "qg" "qk" "pk" "ql"
"qj" "rj" "ri" "si" "rh" "qm" "rk" "rl" "sj" "pm" "ol" "rn" "ro" "qn" "sm" "so" "rp" "sl"
"po" "qo" "qp" "on" "pn" "om" "no" "sp" "rq" "sq" "rr" "oo" "op" "sk" "rj" "fe" "hf" "ag"
"af" "bj" "ah" "ai" "ff" "ea" "ca" "ij" "il" "ad" "cd" "ag" "bf" "ah" "ge" "gd" "gf" "in"
"ip")
sgfc.core=> (print-board (game2board (first (ssimplify (simplify (sgf2tree "resources/kram
ar.sgf"))))))
  A B C D E F G H J K L M N O P Q R S T
1  . . 0 0 X . . . . . . . . X 0 . . .
2  . . 0 X X X . . . . . . . . X 0 0 . .
3  . X 0 0 X 0 X . X X X . . . . X 0 . .
4  X 0 0 X 0 X X X 0 0 0 X . . . . 0 . .
5  . X 0 0 0 X 0 X 0 0 0 X . . X X 0 . .
6  0 0 0 . . 0 0 0 0 X X X X X 0 X 0 . .
7  X 0 0 0 0 0 0 . . 0 0 X X . X 0 0 . .
8  X X X 0 X X X 0 0 0 X X 0 X X 0 . 0 .
9  X X 0 0 X 0 0 X . 0 X X 0 X 0 X X 0 X
10 . X X 0 X 0 X X X 0 0 X 0 X 0 0 0 0 0
11 . . X X X 0 X 0 0 X X X X 0 . 0 X 0 X
12 . . . . X X 0 X 0 0 0 X X 0 0 0 X X X
13 . . . X 0 0 0 0 0 0 X 0 . . X X X . 0
14 . . X . X X X X X 0 . . . X 0 X X .
15 . . . X . . 0 0 X X 0 . . 0 X 0 X 0 X
16 . . . . 0 0 0 X 0 X X 0 0 0 0 0 0 0 X
17 . . X 0 X 0 . 0 X X . X X X 0 . . 0 X
18 . . X 0 X X X 0 0 X . . X . X 0 . 0 .
19 . . . . X 0 0 0 . . . . . . X 0 . . .
nil
sgfc.core=> 
```



# Розвиток системи

- Більше методів для пошуку у структурах даних гри Го
- Підтримка інших настільних ігор
- Підтримка протоколу GTP



# Висновки

В результаті дипломного проектування була створена система, що дозволяє:

- Оброблювати SGF-файли
- Використовувати такі методи пошуку, як:
  - Метод Монте-Карло
  - Альфа-бета відсічення
  - Пошук за базою шаблонів
- Візуалізовувати результат роботи методів пошуку



Дякую за увагу!