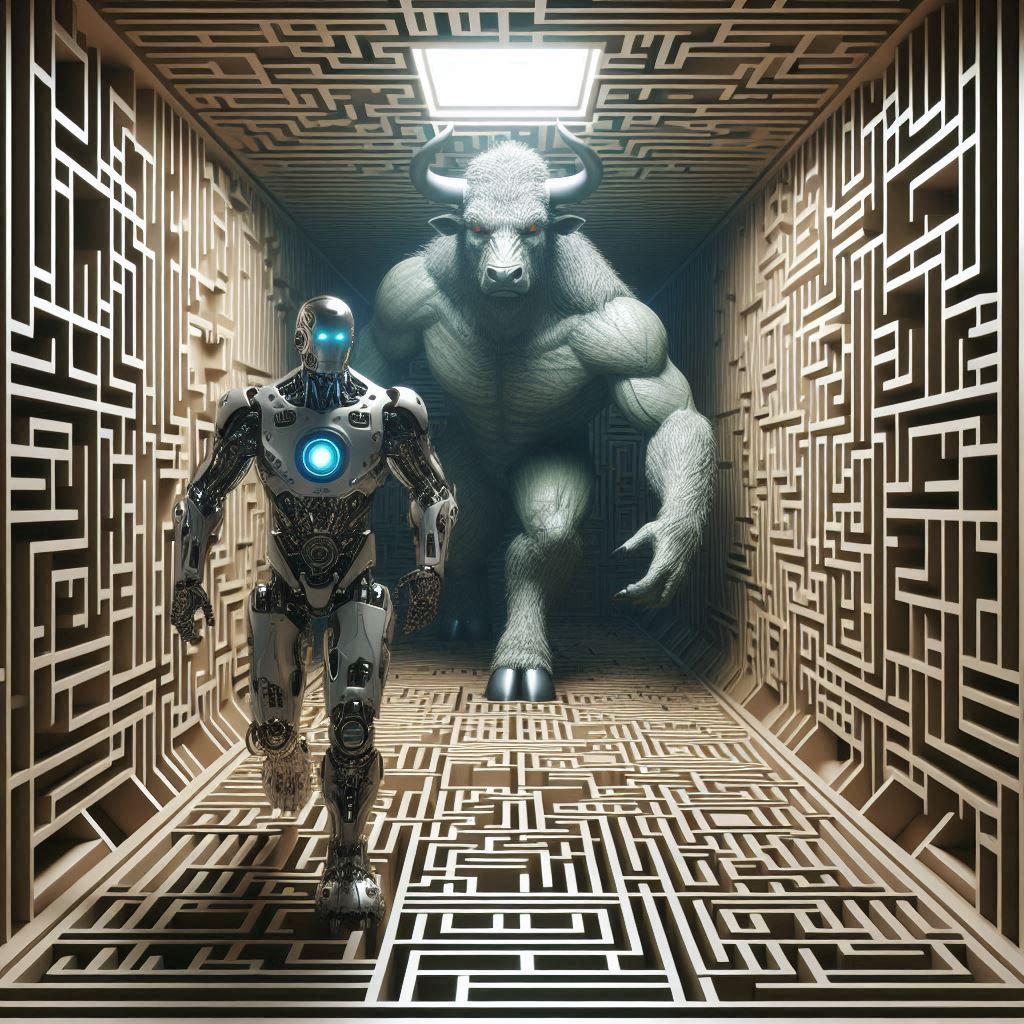
Робот у лавиринту у Кнососу

**Последњи рок за предају пројекта: XX.YY.202Z. 23:59**

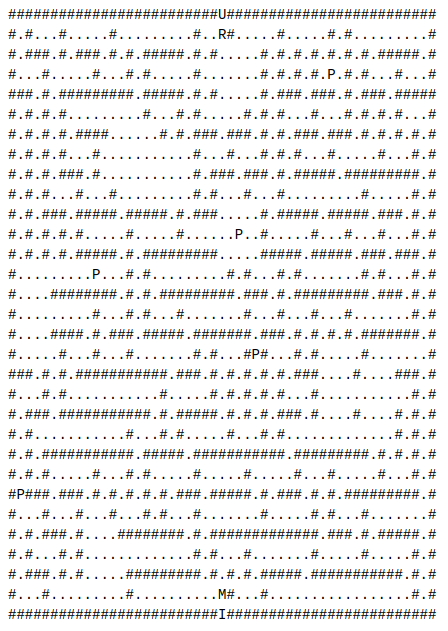
## Опис игре

Лавиринт у Кнососу на острву Крит је био озлоглашена грађевина минојске цивилизације створена од стране једног од највећих изумитеља из античких митова – Дедала. Овај лавиринт је представљао немогућу мисију за тадашњег обичног човека и у њему је по предању страдало много људи. Да ли можете искористити ваше вештине да испрограмирате робота који би успео да пронађе излаз у оваквом лавиринту? Немојте заборавити, у лавиринту није једини изазов био наћи прави пут и избећи ћорсокаке, већ избећи и митско биће Минотаура...



## Ток и правила игре

* На почетку програма потребно је кроз аргументе командне линије задати димензију лавиринта и број предмета са специјалним ефектима. Лавиринт је произвољна матрица где број колона и број редова морају бити веће од 15 (нпр. 20x20, 17x25, 30x20), а број предмета мора бити већи од 3.
* На основу димензије потребно је генерисати поставку на следећи начин:
  + ‘#’ - зидови
    - спољашњи зидови морају попунити димензије матрице
    - блокове унутрашњих зидова треба насумично распоредити, где је број блокова двострука вредност димензије лавиринта, а једини услов је да буду постављени тако да је пролаз могућ
  + ‘U’ - улаз насумично распоређен на првом реду лавиринта
  + ‘R’ – робот распоређен на поље које је тачно један ред испод улаза
  + ‘I’ – излаз насумично распоређен на последњем реду лавиринта
  + ‘.’ - пролаз представљен једним карактером размака, насумично распоређен тако да омогући кретање
  + ‘M’ - Минотаур, насумично распоређен са јединим условом да буде могућ пролазак поред њега
  + ‘P’ - предмет који нуди одређени ефекат (укупан број предмета је задат аругментом командне линије)
* Затим, следи интерактивни део игре:
  + Кретање робота
    - корисник уносом са тастатуре врши кретање робота (горе, доле, лево и десно)
    - команде за кретање су произвољне (могу бити стрелице или по жељи дефинисани карактери)
    - Робот не може да прође кроз зид
  + Кретање минотаура:
    - насумично померање за једно поље (лево, десно, горе или доле)
    - уколико се робот нађе на једно поље од Минотаура, Минотаур неће играти насумично већ ће појести робота
    - Минотаур не може да прође кроз зид
    - уколико стане на поље на ком је предмет, Минотаур тај предмет уништава
  + Предмети и њихови специјални ефекти:
    - Специјални ефекти сваког предмета трају 3 потеза
    - Када робот стане на предмет, насумично се генерише ефекат који предмет доноси
    - ‘Магла рата’ - смањује кориснику видљивост на подматрицу димензије 3x3
    - ‘Мач’ - овим предметом је могуће уништити Минотаура
    - ‘Штит’ - могућност одбране од Минотаура
    - ‘Чекић’ - омогућава пролазак кроз зид



Пример генерисаног лавиринта дименција 51x31.

## Функционални захтеви

Написати Це++ програм који омогућује следеће:

* Учитавање димензије лавиринта и броја предмета као аргумената командне линије
* Динамичко креирање лавиринта са свим задатим елементима
* Алгоритам који прави поставку лавиринта на основу задатих димензија имплементирати у засебној библитеци
* Мерење времена потребног за генерисање табеле лавиринта
* Интеракција кроз конзолу
* Сва правила игре омогућена и испоштована
* Коришћење наслеђивања приликом прављења предмета и њихових специјалних ефеката
* На крају игре изгенерисати датотеку у којој ће бити приказан исход и слика лавиринта након последњег потеза. Крај игре подразумева:
  + команду са тастатуре (слово ‘Q’) за гашење програма
  + Долазак робота на финиш
  + Минотаур је уништио робота
* Тестирање кода у слободној форми (сви случаји са којима је тестирано решење морају бити наведени у документацији)
* Ток игре и руковање датотекама треба имплементирати као одвојене класе

**НАПОМЕНЕ:**

* Обратити пажњу на употребу адекватних структура података
* Водити рачуна о динамичким структурама података и правилном ослобађању ресурса
* Програм поделити у смислене целине и класе
* Проверити исправност програма на различитим тестним случајевима
* Додатни савет: Уколико решење не поседује генерисање лавиринта или коришћење засебне библиотеке, користити ручно направљену таблицу лавиринта како би показали исправност рада остатка програма (сваки део програма се оцењује засебно!)

## Стил кодовања

* + - * У сваком заглављу и модулу са изворним кодом додати кратак опис функционалности, информацију о ауторима, и датум и аутора последње измене.
      * Коментарисати најбитније слободне функције, функције чланице и атрибуте класа. За функције обезбедити кратак опис функционалности, листу улазних аргумената, повртану вредност и уколико користи тип изузетка.
      * Обратити пажњу на индентацију и форматирање кода, на стил и формат именовања промељивих и функција, као и на дужине линија.
      * Код треба да буде прегледан, читљив и да садржи корисне коментаре.
      * Више о начину и стилу кодовања доступно је у следећем документу: <http://www.stroustrup.com/Programming/PPP-style-rev3.pdf>

# Излазни резултати

Одабиром релевантних тестних случајева у документацији приложити дискусију о промени времена потребног за креирање лавиринта у зависности од димензије лавиринта.

# Извештај

Треба да садржи следеће:

* + - * Насловну страну са информацијама о аутору.
      * Опис и/или анализу:
        + Рада У/И подсистема (учитавање и испис/упис).
        + Списак свих класа, изузетака и слободних функција.
        + Објашњење најбитних атрибута класа и функција чланица, слободних функција и изузетака.
        + Структуре аргумената командне линије и пример коришћења.
        + Структуре излазне датотеке.
        + Напреднијих ОО концепата (уколико су коришћени): наслеђивање, преклапање оператора, полиморфизам, шаблони итд.
        + Тестних случајева
        + Ставке специфициране у претходној секцији „Излазни резултати“.
        + Уочени проблеми и ограничења.

# Упутство за предају пројекта

* + - Пројекат треба да буде архивиран и именован на следећи начин:

**Indeks\_Ime\_Prezime.zip**

(пример: SW123\_Petar\_Petrovic.zip)

* + - Архиву окачити на канвас у предвиђеном временском року.
    - Архива треба да садржи следеће директоријуме:
      * **Kodovi** – садржи датотеке са изворним кодом (.cpp и .h/.hpp) и пројектне датотеке (.sln, vcxproj и filters).
      * **Testovi** – садржи све тестне датотеке. **(опционо)**
      * **Dokumentacija** – садржи пројектну документацију *Izvestaj.doc*.
    - Архива **НЕ СМЕ** садржати следеће датотеке:

.sdf

.suo

.user

.obj

.lib

.exe

ipch

Debug директоријум

Release директоријум

* + - * Приложен је референтни пример решења пројектног задатка.
    - На термин одбране потребно је **донети штампану верзију документације**.