

Звіт

З дисципліни «Інтелектуальні системи»

Лабораторна робота №6

«PacMan PLUS»

Виконали:

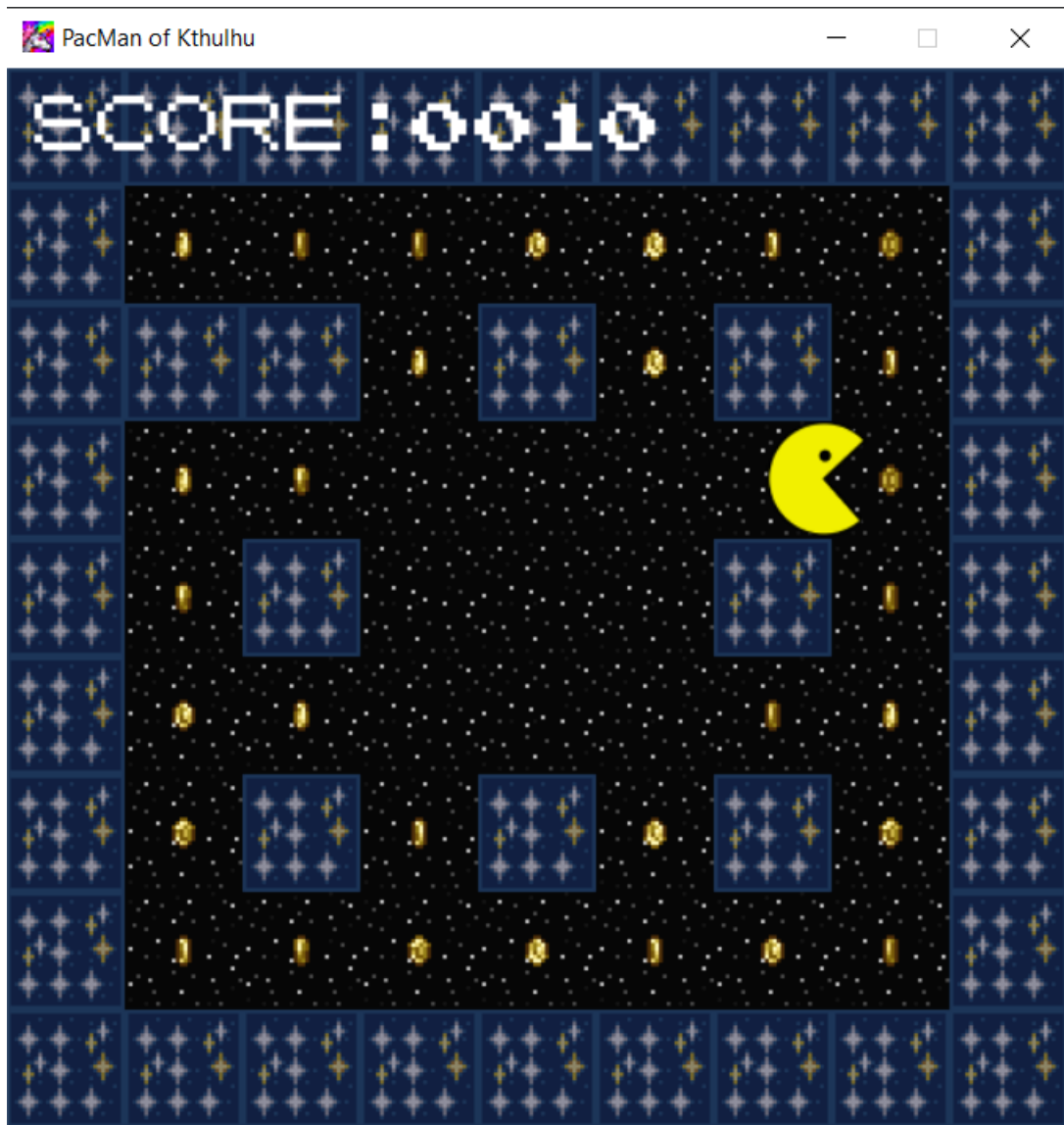
Давиденко Олексій

Макаревич Олександр

Шабліовська Діана

## 1. Рахунок і монети (Діана Шабліовська)

По всьому рівню, в межах коридорів, за виключенням стартової зони в центрі, розсипаються монети, які Пакман хоче з'їсти.



За кожну з'їдену монету Пакман отримує 10 очків. Як тільки всі монети в рівні з'їдені, Пакман виграє рівень. Рахунок очків зберігається між рівнями.

## 2. Привиди (Олександр Макаревич)

Привиди можуть з'являтися на рівні в будь-якій точці, віддаленій від центру. Вони рухаються зі сталою швидкістю. З кожним своїм рухом привид може або наблизитися до Пакмана, слідуючи найкоротшим шляхом, або піти в випадковому напрямку.

```

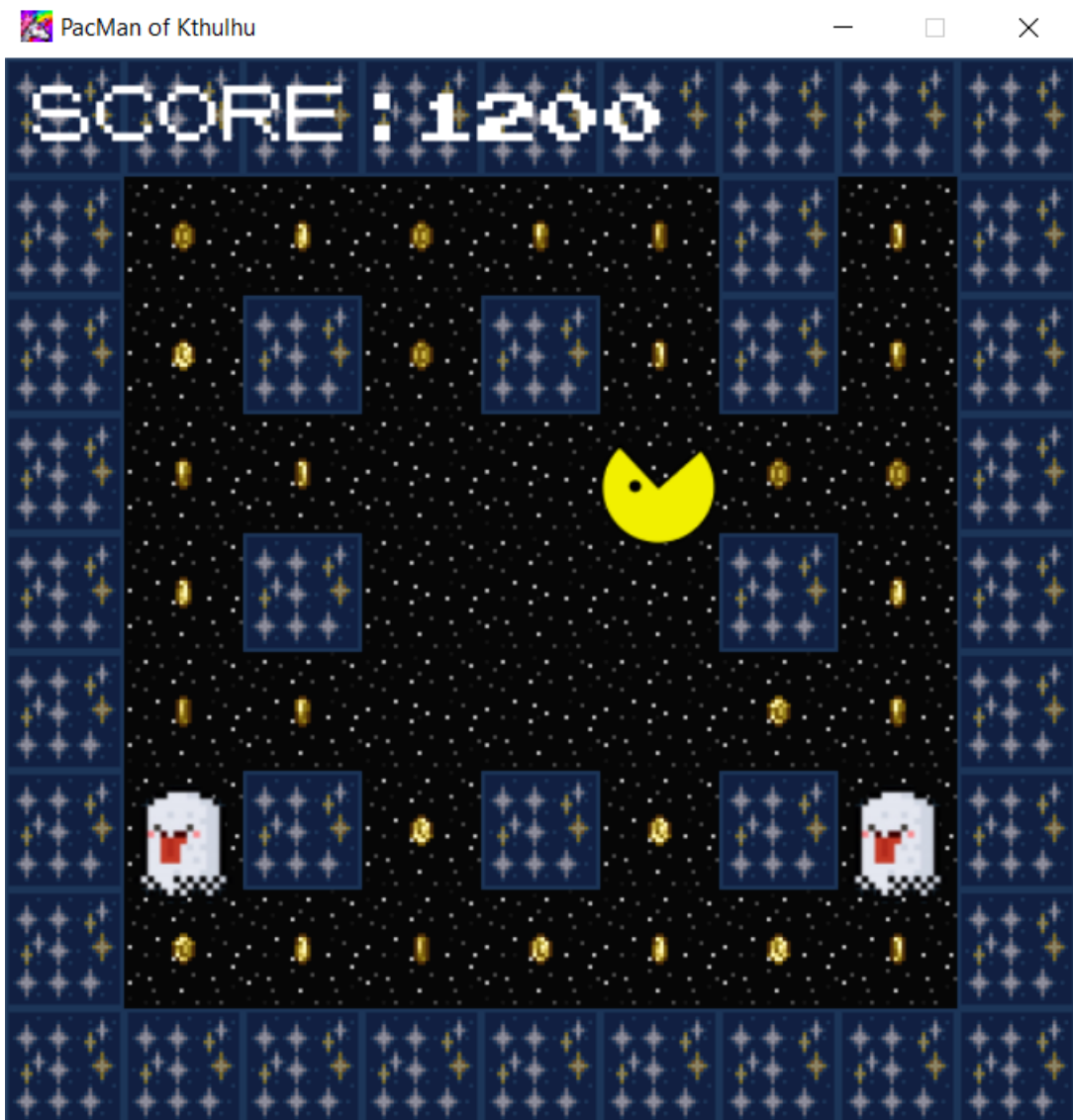
class Ghost(Character):
    def __init__(self, tile_x, tile_y, move_frames, random_move_chance):
        # --- #

    def update(self, level, pacman):
        # --- #

    def pick_move(self, level, pacman):
        if np.random.random_sample() < self.random_move_chance:
            # pick random move
            directions = [(0, 1), (0, -1), (-1, 0), (1, 0)]
            possible_moves = []
            for direction in directions:
                target_y = self.curr_tile_y + direction[0]
                target_x = self.curr_tile_x + direction[1]
                if level.tile_map[target_y, target_x] == 0:
                    possible_moves.append(direction)
            return random.choice(possible_moves)
        else: # pursue pacman
            pacman_x, pacman_y = pacman.curr_tile_x, pacman.curr_tile_y
            moves_to_pacman = level.find_shortest_path(self.curr_tile_x,
self.curr_tile_y,
                                                    pacman_x, pacman_y)

            if moves_to_pacman:
                return moves_to_pacman[0]

```



### 3. Рівні складності (Олексій Давиденко)

Рівень складності для певного рівня визначається рядом параметрів:

- *ghost\_frames\_per\_tile* – час, який потрібен приви́ду для того щоб подолати одну клітинку. Обернено пропорційний швидкості
- *ghost\_amount* – кількість приви́дів на рівні
- *random\_move\_chance* – вірогідність того що привид зробить випадковий крок замість того щоб слідувати за Пакманом

```
# parameters that dictate how hard the game becomes at each difficulty level
difficulty_settings = {
  0: {"ghost_frames_per_tile": 99, "ghost_amount": 0, "random_move_chance": 0.0},
  1: {"ghost_frames_per_tile": 40, "ghost_amount": 1, "random_move_chance": 1.0},
  2: {"ghost_frames_per_tile": 40, "ghost_amount": 1, "random_move_chance": 0.1},
  3: {"ghost_frames_per_tile": 30, "ghost_amount": 1, "random_move_chance": 0.3},
  4: {"ghost_frames_per_tile": 40, "ghost_amount": 2, "random_move_chance": 0.4},
  5: {"ghost_frames_per_tile": 40, "ghost_amount": 2, "random_move_chance": 0.1},
  6: {"ghost_frames_per_tile": 40, "ghost_amount": 3, "random_move_chance": 0.4},
  7: {"ghost_frames_per_tile": 30, "ghost_amount": 4, "random_move_chance": 0.1} #
  # this is basically impossible
}
```



#### 4. Штучний інтелект Пакмана (Олексій Давиденко)

Для визначення найкращого кроку (вправо, вліво, вгору або вниз) Пакман моделює дерево ігрових станів. Розгалудження відбуваються тоді, коли у Пакмана є вибір в напрямку руху. В якості оптимізації, Пакману дозволяється повертатися назад тільки у випадку, коли він підібрав своїм рухом монету. Це значно звужує дерево пошуку.

На кожному вузлі дерева зберігається промодельований стан рівня. Пакман припускає, що привиди рухатимуться до нього по найближчому шляху (за принципом мін-максної гри). Привиди при цьому не кооперуються і грають незалежно. У Пакмана є певна «глибина» прорахунку стратегій, обмежена розрахунковою потужністю ПК.

