DISCUSSION POINTS — Лекция 2

Авторское мышление • NumPy (модель) + Pygame (вид)

1) Data representations: Scalar · Vector · Matrix

• Скаляр — единичное число (напр., TILE_SIZE=32, FPS=60).
• Вектор — упорядоченная тройка RGB (цвет).
• Матрица — screen_matrix[y, x] ∈ {0..5} — индексы палитры.
• Конвейер: (y, x) → index → RGB → тайл на экране.

2) Model ↔ View: NumPy (data) и Pygame (rendering)

• Модель: screen_matrix и палитра (NumPy) — источник истины.
• Вид: Pygame читает модель и рисует; порядок важен: тайлы → сетка.
• Рендер не меняет модель; изменения делаем срезами/масками.

3) Matrix traversal и единое правило отображения

• Обход построчно: сначала y (строка), затем x (столбец).
• Единое правило: (y, x) ↦ (x*TILE_SIZE, y*TILE_SIZE, TILE_SIZE, TILE_SIZE).
• Одна формула убирает путаницу осей и упрощает проверку.

4) Намеренная неэффективность (и оценка сложности)

• Полный перерендер 16×32 клеток на кадр: ~O(TILE_Y*TILE_X).
• Педагогически прозрачно: Модель → Вид видно «в лоб».
• Дальше можно показать апгрейды: dirty rects, blit тайлов, surfarray.

5) Авторское мышление (позиция + ответственность + шаг вперёд)

• Ученик формулирует свои правила (срезы/маски) → чувствует себя автором.
• Ответственность: понимаю последствия правил и стою за результат.
• Движение вперёд: фиксирую версию и делаю следующий осознанный шаг.

Мини-рубрика (на урок):

• Оригинальность правила: что здесь моё?
• Воспроизводимость: по описанию можно повторить.
• Шаг вперёд: что изменю дальше и почему.