УРСУЛА
МАСТЕРСКАЯ ИГР

# Обмен данными между объектами в игре Ink → ПРИМС через глобальные переменные



# Два метода интеграции Ink и ПРИМС

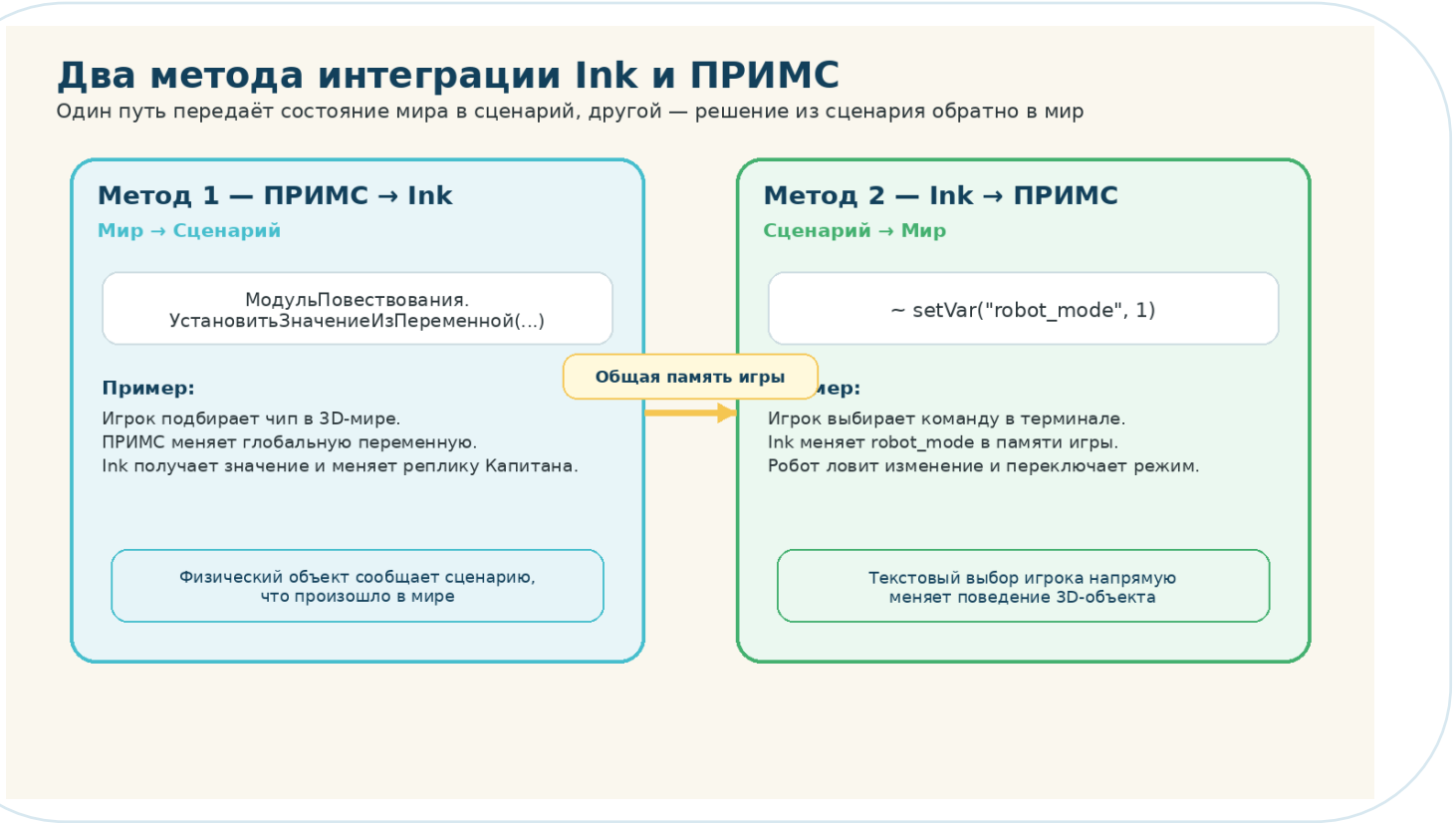
Сравниваем передачу данных «мир → сценарий» и «сценарий → мир»

**1** **ПРИМС → Ink**
Физическое действие меняет реплику

**2** **Ink → ПРИМС**
Выбор в диалоге меняет физический мир

**3** **Общая память**
Оба метода работают через глобальные переменные

**Сегодня осваиваем обратный путь: выбор в Ink переключает режим удалённого робота**



# Мост в коде сценария

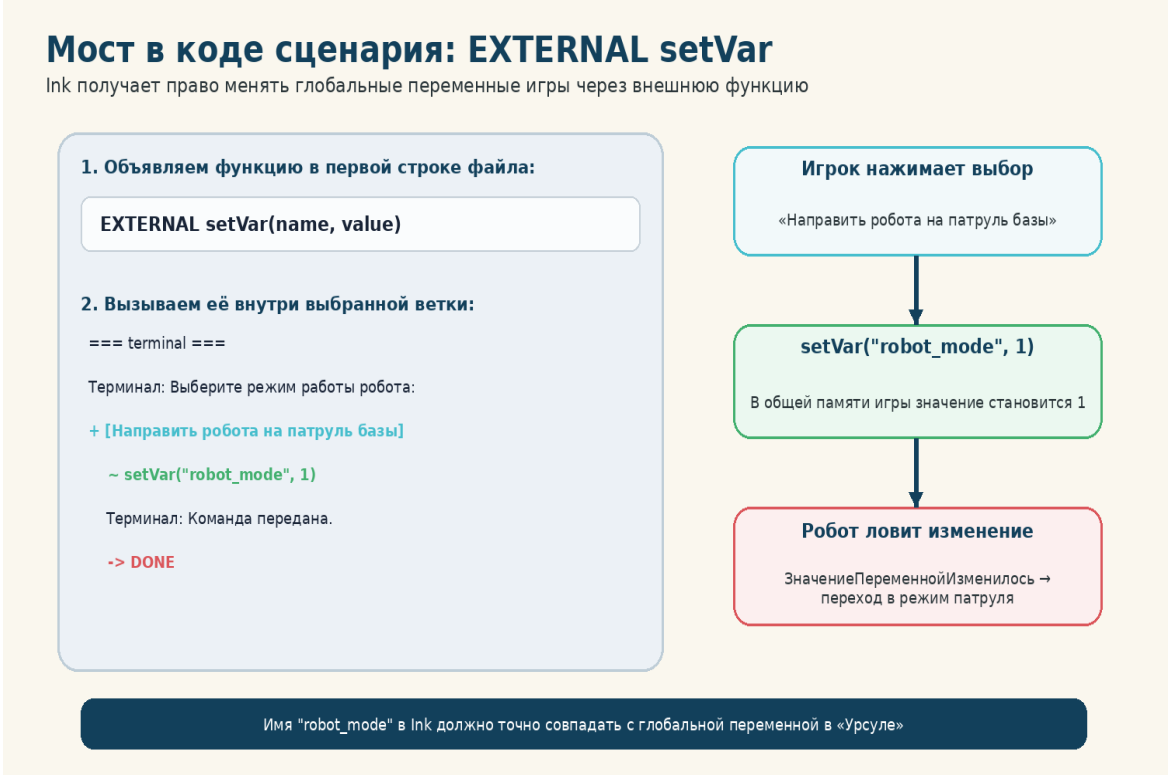
EXTERNAL setVar разрешает Ink менять глобальные переменные игры

**1 Объявление**
EXTERNAL setVar(name, value) — в первой строке файла

**2 Вызов**
~ setVar("robot_mode", 1) — внутри выбранной ветки

**3 Результат**
Робот получает событие изменения переменной и меняет режим

**Имя переменной в Ink должно точно совпадать с именем глобальной переменной в «Урсуле»**



# Простое устройство терминала

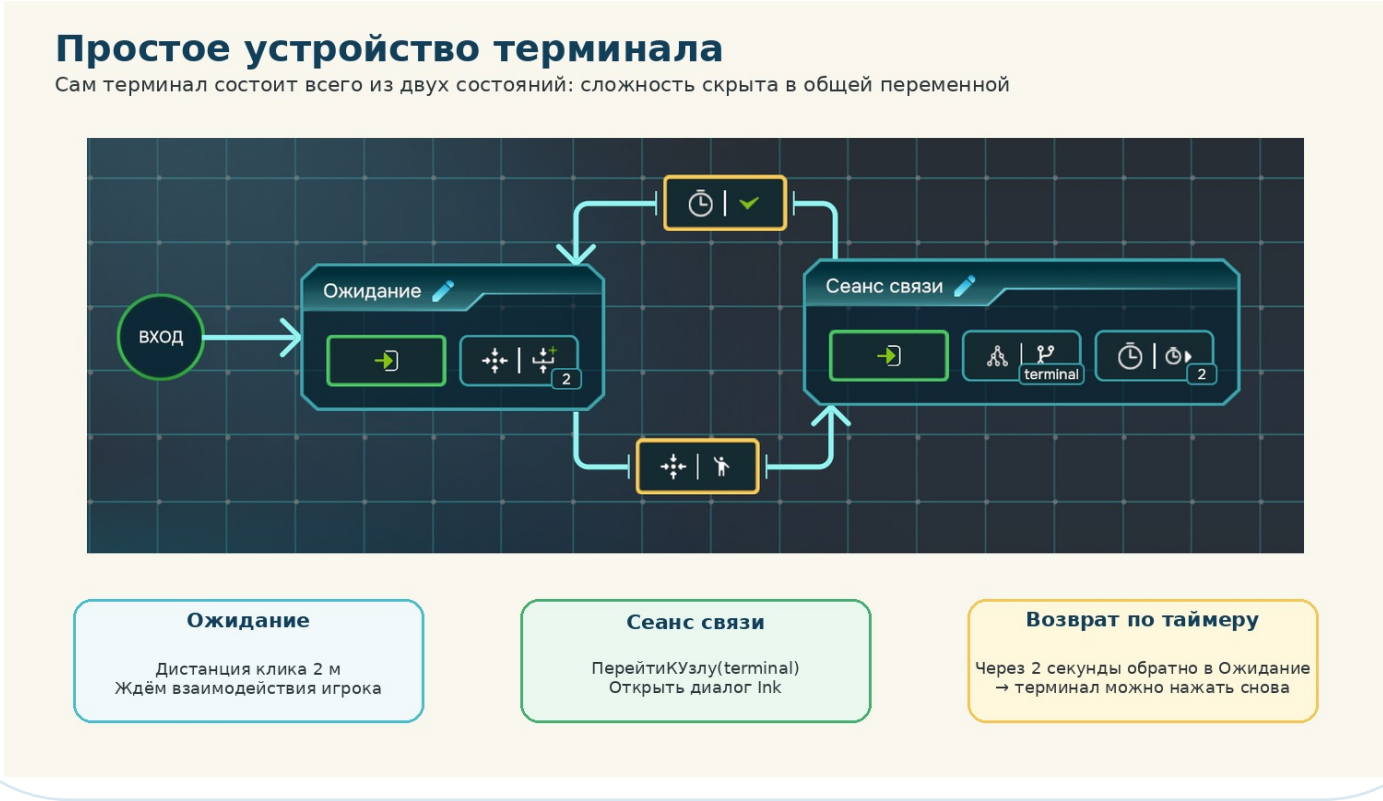
Два состояния: ожидание игрока и короткий сеанс связи

**1 Ожидание**
Дистанция клика 2 м, ждём взаимодействия

**2 Сеанс связи**
Открываем узел terminal через МодульПовествования

**3 Возврат**
Через 2 секунды таймер возвращает терминал в ожидание

**Терминал простой: вся связь с удалённым роботом происходит через robot_mode**



# Собираем всю систему

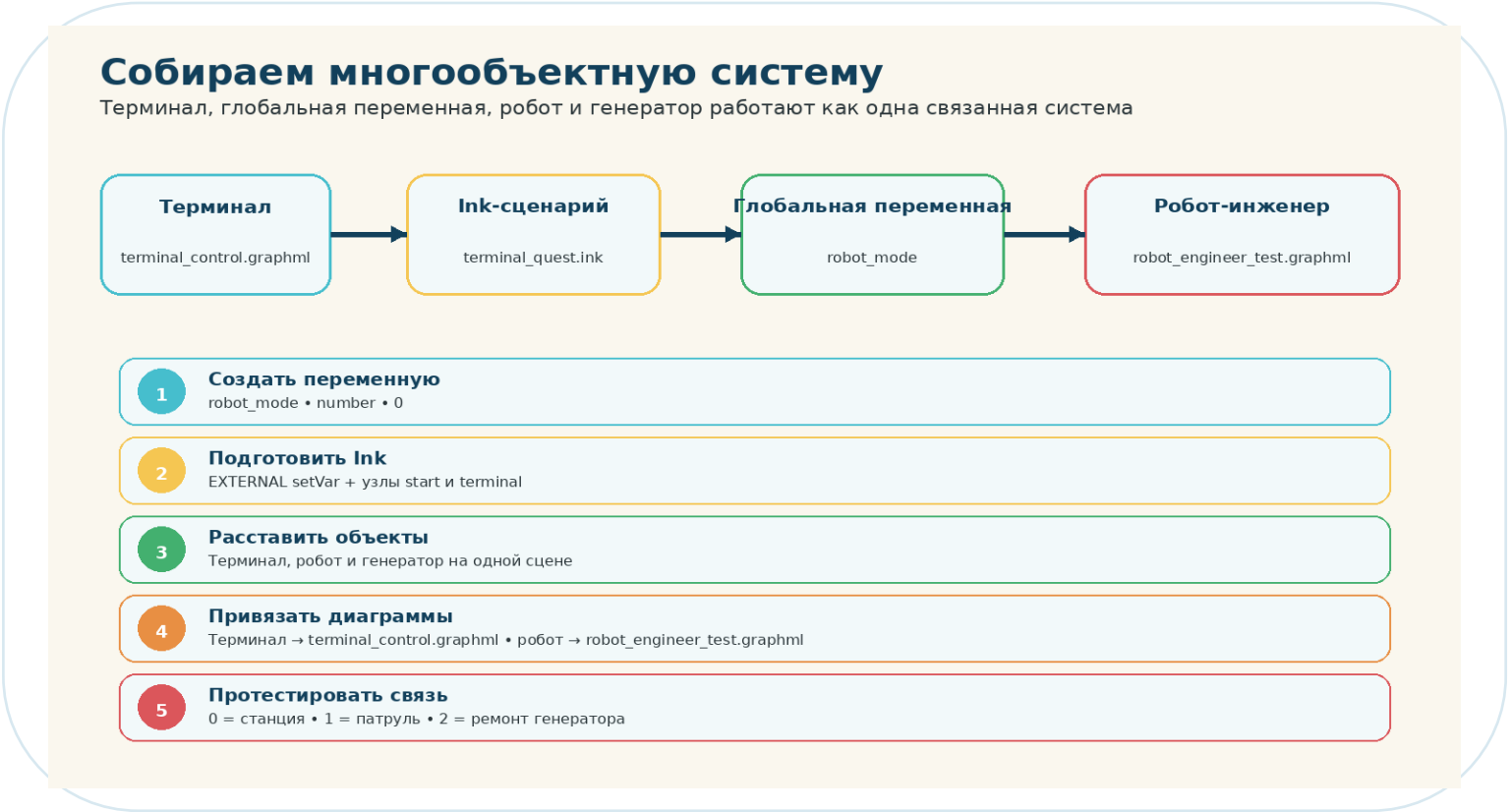
Терминал + Ink + глобальная переменная + робот-инженер

**1 Переменная**
Создать robot_mode типа number со значением 0

**2 Объекты**
Разместить терминал, робота и generator на сцене

**3 Тест**
Проверить режимы 0, 1 и 2 через команды терминала

**Цель: выбор в терминале должен физически переключать удалённого робота**



# Итоги: связанный игровой мир

Глобальные переменные соединяют сценарий, интерфейс и физические объекты

**1** **ПРИМС → Ink**

Мир сообщает сценарию о событиях

**2** **Ink → ПРИМС**

Сценарный выбор меняет состояние мира

**3** **Многообъектность**

Несколько объектов работают через одну общую память

**Теперь терминал может управлять роботом на другом конце карты через robot_mode**

