Interakcja człowiek–system w XR

WiRR · Wykład 05 · Automatyka i Robotyka · 3 ECTS · 2026/2027 · v1.3.19

dr inż. Mateusz Pomianek

Katedra Informatyki i Automatyki, Politechnika Rzeszowska

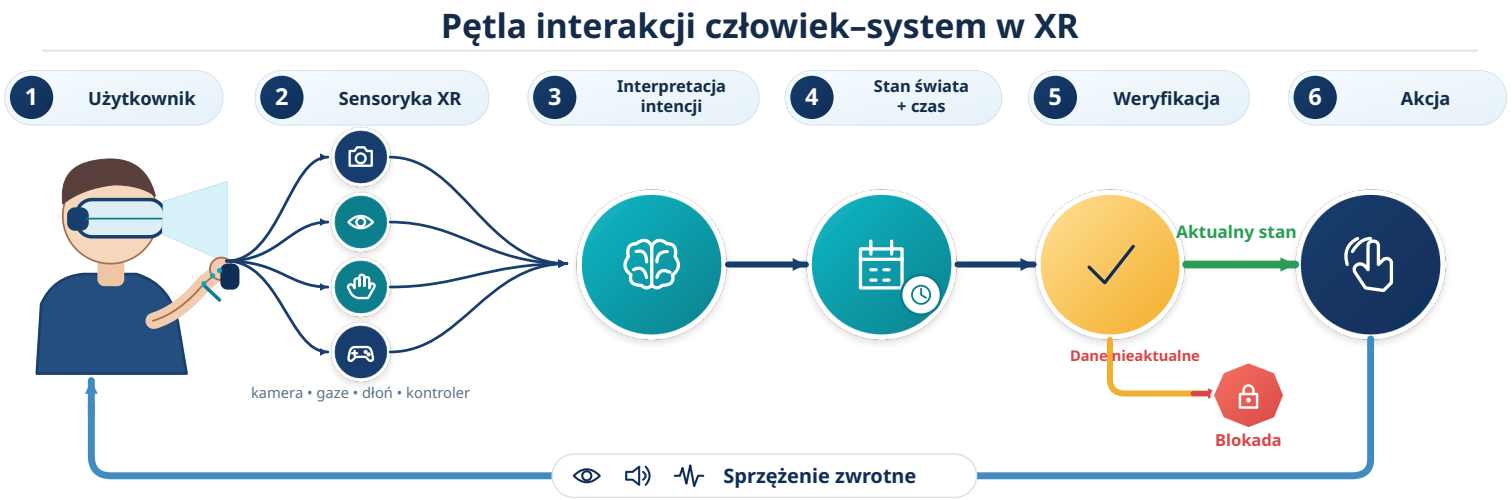
27.10.2026

Jak wybrać obiekt bez przypadkowego innego polecenia



- **Dobieramy** modalność do zasięgu i rozmiaru celu
- **Oddzielamy** wskazanie, wybór oraz wykonanie akcji
- **Oceniamy** szybkość, błędy i wysiłek w jednym zadaniu

Interakcja jako pętla sprzężenia zwrotnego



Bezpieczeństwo interakcji: wskazanie obiektu nie jest równoznaczne z wykonaniem polecenia. Utrata celu lub zbyt mała pewność musi umożliwiać anulowanie.

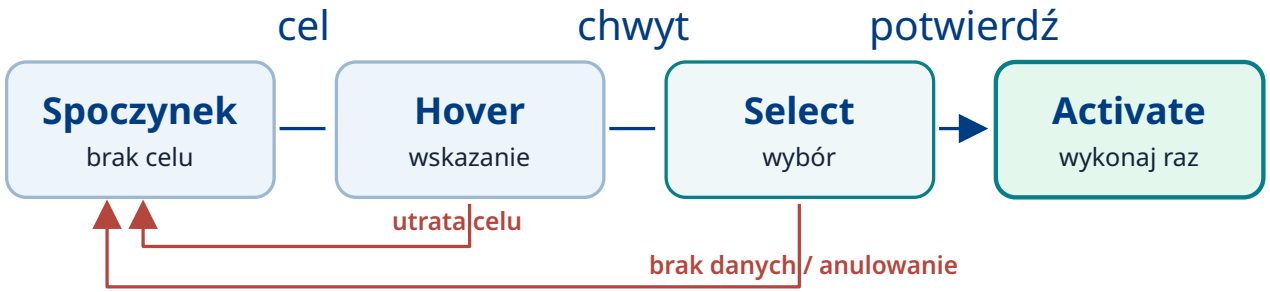
Hover, select i activate oznaczają różne zdarzenia



Osobne zdarzenia umożliwiają potwierdzenie i anulowanie.

Podświetlenie celu nie powinno samo uruchamiać kosztownej operacji

Interakcja jako automat stanów: wskaż, wybierz, wykonaj, anuluj



Wskazanie nie może samo wykonać polecenia

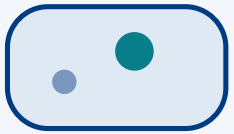
Aplikacja dopuszcza aktywację dopiero po kontroli stanu oraz aktualności pozy

- `HoverEnter`: wskaż; `SelectEnter`: wybierz.
- `Activate`: zweryfikuj warunki, wykonaj **raz**.
- `HoverExit` lub brak aktualnych danych: odrzuć operację.

Logika wykonania: sprawdź `CanExecute()`, ustaw `consumed = true` **przed** wywołaniem `onAccepted`. Po odrzuceniu wymagaj nowego wyboru. Pełna klasa C# z obsługą zdarzeń

Kontrolery, a śledzenie dłoni

Kontroler i przycisk



Wyraźne zdarzenie

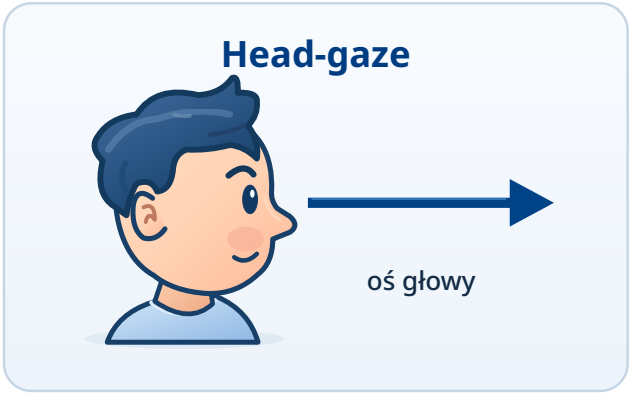
Śledzenie dłoni



Okluzja i niepewność

Kontrolery	Śledzenie dłoni
Przyciski zapewniają wyraźne potwierdzenie	Nie wymaga trzymania urządzenia
Dobre dla precyzyjnych poleceń	Wrażliwe na przesłonięcia i jakość obrazu
Zajmują dłonie	Rozpoznanie gestu ma niepewność

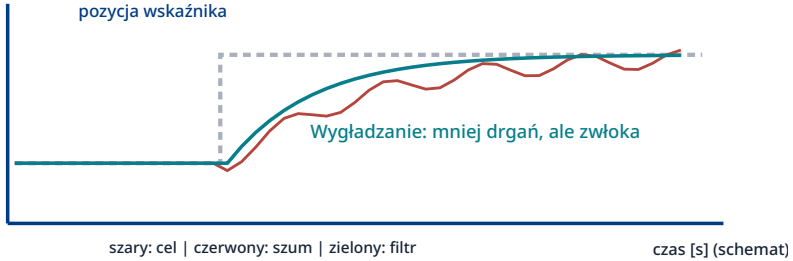
Head-gaze, a eye tracking





Head-gaze wyznaczamy z orientacji głowy. **Eye tracking** estymuje kierunek spojrzenia oczu. Użytkownik może poruszać oczami bez obrotu głowy, więc tych pomiarów nie należy utożsamiać.

Drżenie wskaźnika, stabilność kontra opóźnienie

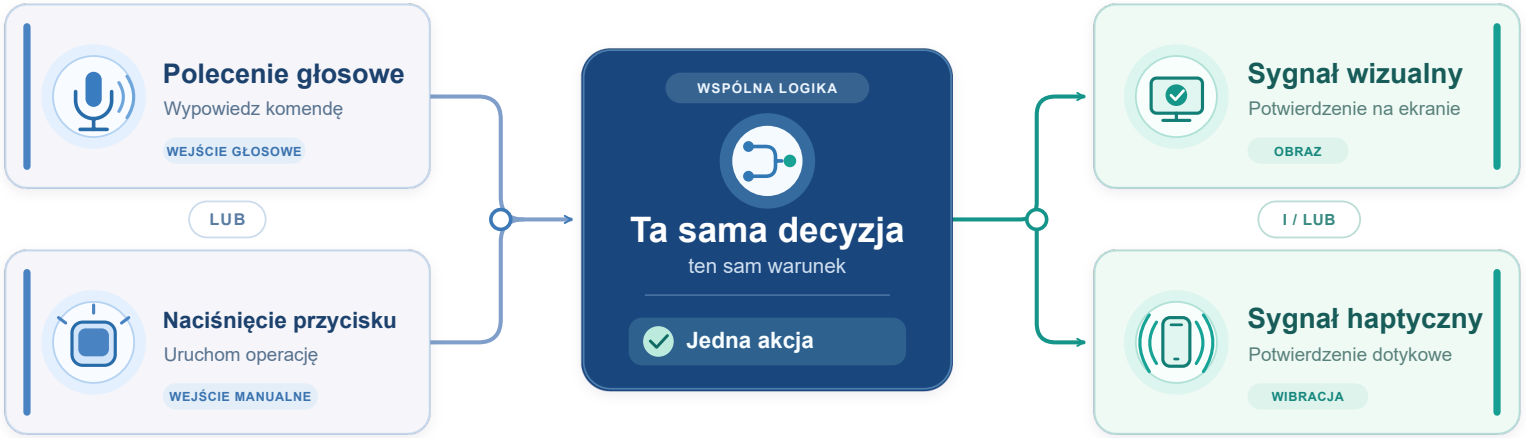


Ruch dłoni i szum śledzenia powodują drgania promienia. Wygładzanie zmniejsza jitter, ale zwiększa opóźnienie reakcji.

```
// Metoda komponentu Unity (using UnityEngine)
static Vector3 Smooth(Vector3 current, Vector3 target, float dt, float tau)
{
    float safeTau = Mathf.Max(tau, 0.001f);
    float a = 1f - Mathf.Exp(-Mathf.Max(dt, 0f) / safeTau);
    return Vector3.Lerp(current, target, a);
}
```

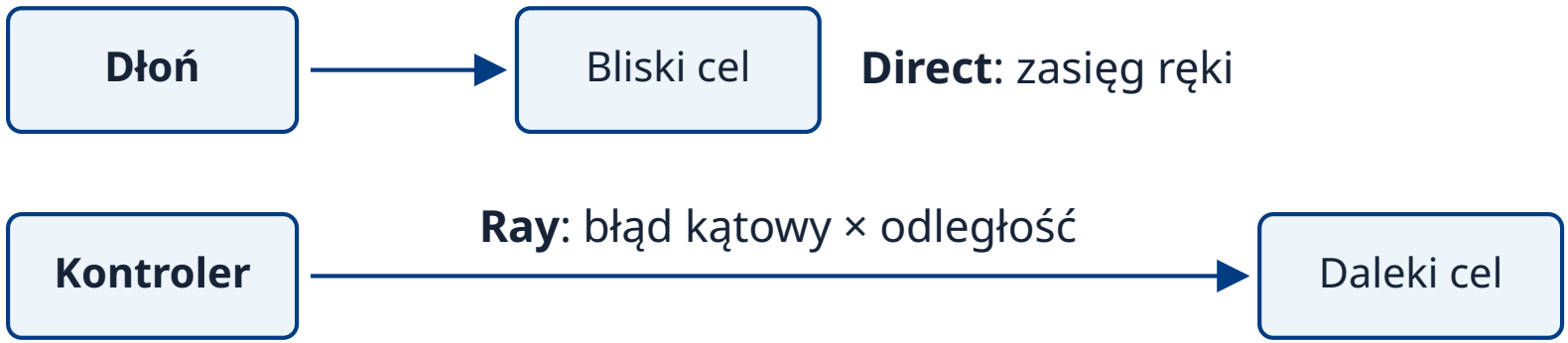
Wywołanie w `Update()`: `cursor = Smooth(cursor, target, Time.unscaledDeltaTime, tau);`

Głos i haptyka



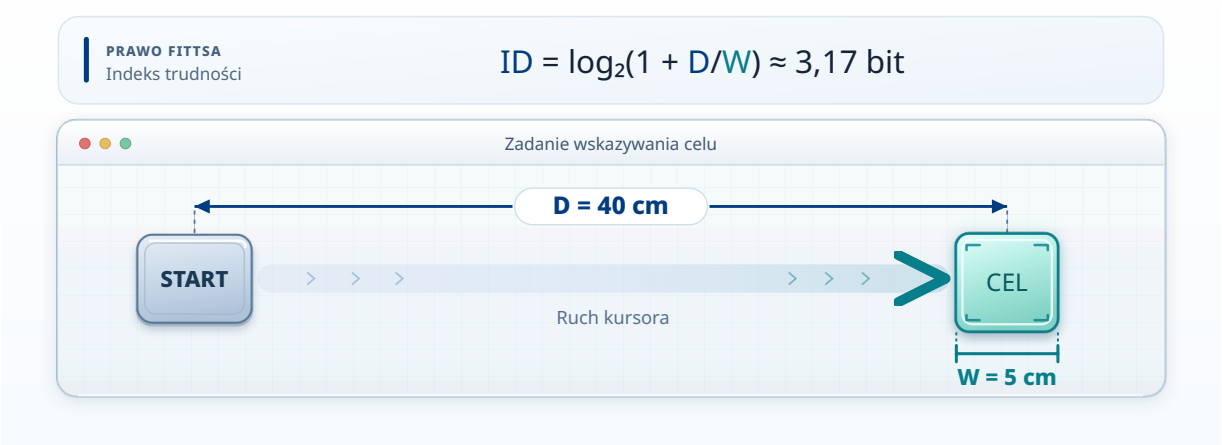
Głos: wygodny bez użycia rąk, lecz zależny od szumu i jakości rozpoznania **Haptyka**: potwierdza stan przez wibrację, ale nie dowodzi wykonania komendy

Direct interaction i ray interaction



Interakcja bezpośrednia sprawdza się w pobliżu obiektu, promieniowa umożliwia wskazywanie odległych celów

Prawo Fittsa



Dla wskazywania celu czas ruchu można modelować zależnością:

$$MT = a + b \log_2(1 + D/W)$$

gdzie D - odległość, W - szerokość celu w kierunku ruchu, a, b - są parametrami modelu (wyznaczamy je empirycznie dla konkretnej techniki wskazywania)

Trudność celu: obliczenie bez dopasowania czasu

Dla $D = 40$ cm i $W = 5$ cm:

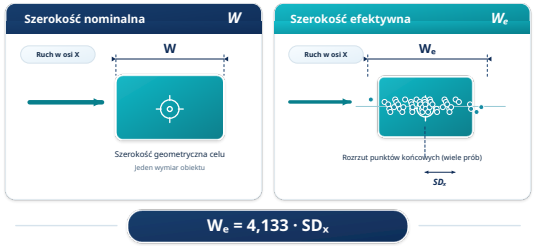
$$ID = \log_2(1 + 40/5) \approx 3,17 \text{ bit}$$

Dla $W = 2,5$ cm: $ID \approx 4,09$ bit

Bez wyznaczenia współczynników a i b nie znamy czasu MT.

Nie przenosimy ich automatycznie między myszą, promieniem i spojrzeniem.

Szerokość nominalna i efektywna to różne dane



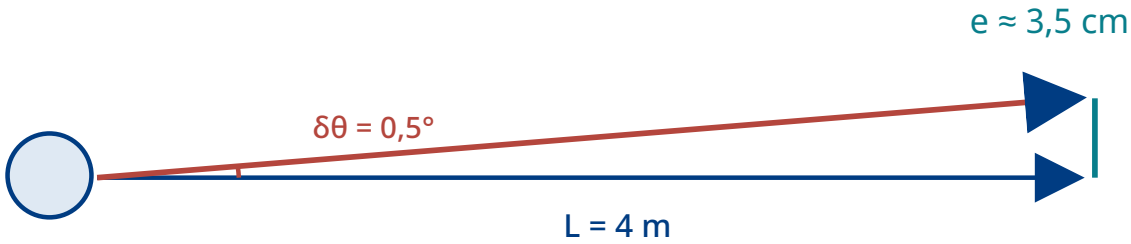
W prawie Fittsa `W` to nominalna szerokość celu w kierunku ruchu. Efektywną szerokość można oszacować na podstawie rozrzutu punktów zakończenia ruchu przy określonej metodzie analizy.

Szerokość nominalna `W`: geometria celu w kierunku ruchu. **Szerokość efektywna `W_e`:** rozrzut punktów zakończenia wielu prób.

Przy określonym protokole badawczym można przyjąć $W_e = 4,133\,SD_x$, gdzie SD_x to odchylenie standardowe punktów końcowych wzdłuż osi ruchu.

Wymaga to wielu prób, a nie samego odczytu rozmiaru przycisku. Wskazania promieniem i gaze należy analizować oddzielnie.

Błąd kątowy staje się błędem bocznym



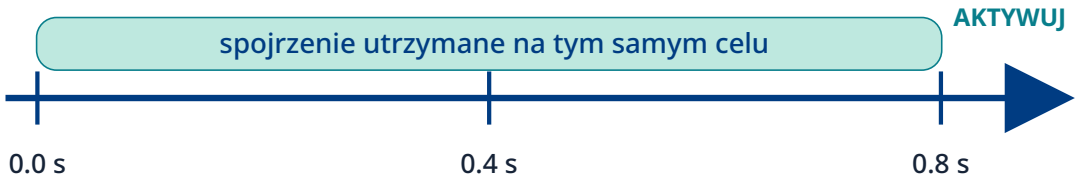
e = L · tan(δθ) • model geometryczny

$$e = L \tan(\delta\theta)$$

Dla $L = 4$ m i $\delta\theta = 0,5°$: $e \approx 3,5$ cm

Przy odległym małym celu można zwiększyć obszar wyboru lub dodać wspomaganie wskazywania i osobne potwierdzenie

Dwell: aktywacja przez spojrzenie

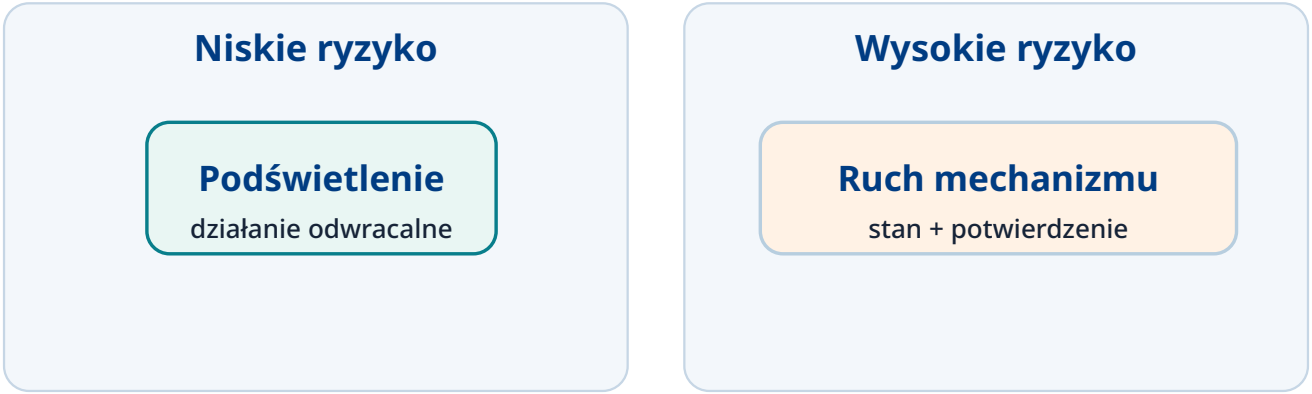


GazeExit przed progiem → licznik = 0

Dwell: aktywacja po utrzymaniu spojrzenia na **jednym celu**. Przykładowy próg `0,8 s` nie jest normą: krótszy sprzyja niezamierzonym aktywacjom, dłuższy wydłuża wybór.

Po `GazeExit` licznik się zeruje; kolejne `GazeEnter` może ponownie rozpocząć pomiar. Jeden ciągły okres spojrzenia powoduje **co najwyżej jedną aktywację**.

Potwierdzenie i anulowanie zależą od ryzyka działania



Ryzyko wyznacza warunki, nie samą modalność

Niski koszt błędu: podświetlenie lub przestawienie koloru **Wysoki koszt błędu:** ruch robota, usunięcie danych, odblokowanie mechanizmu

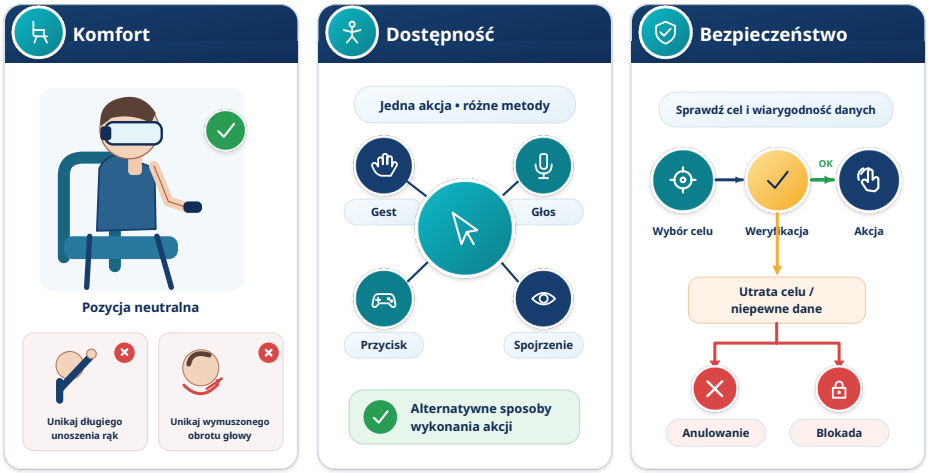
W drugim przypadku wymagamy świadomego potwierdzenia, możliwości anulowania i sprawdzenia aktualnego stanu.

5 minut przerwy

Pytania do samodzielnego powtórzenia

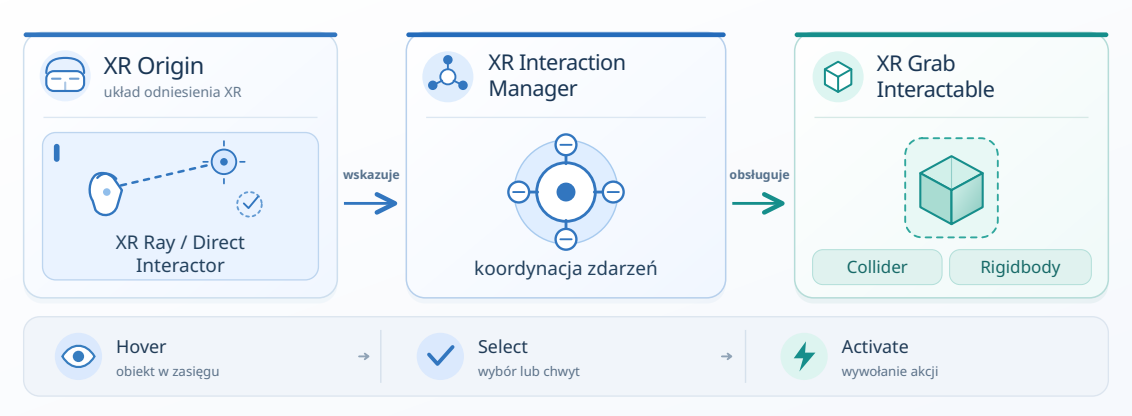
1. Jak odróżnić hover, select i activate przy sterowaniu kontrolerem lub gestem?
2. Jak rozmiar i odległość celu wpływają na trudność w prawie Fittsa?
3. Czym różnią się head-gaze i eye tracking oraz jak uniknąć Midas Touch?
4. Kiedy wiek danych lub brak potwierdzenia powinien zablokować aktywację?

Komfort, dostępność i bezpieczeństwo



- **Komfort**: ograniczaj długie unoszenie ramion i wymuszony obrót głowy
- **Dostępność**: zapewnij alternatywę dla gestów i poleceń głosowych
- **Bezpieczeństwo**: wymuś anulowanie przy utracie celu lub danych

Unity: komponenty XRI i anulowanie dwell



Komponenty: `XR Origin`, `XR Interaction Manager`, `XR Ray/Direct Interactor` i `XR Grab Interactable` z `Collider` oraz `Rigidbody`.

Zdarzenia: `Hover`, `Select` i `Activate` podłącz do metod bramki. `GazeExit()` zeruje `gazing`, `activated` i `gazeStart`.

Ograniczenie: DwellSelector symuluje sygnał gaze, **nie mierzy spojrzenia oczu**.

Test A/B: jedna zmiana, ten sam cel

A: wskazywanie bezpośrednie. **B:** wskazywanie promieniem. Identyczne cele, odległości i reguły potwierdzenia.

```
private float trialStart;
private string condition;

void BeginTrial(string mode) {
    condition = mode;                // A lub B
    trialStart = Time.unscaledTime; // pokazanie celu
}
void ConfirmTrial() {
    float elapsed = Time.unscaledTime - trialStart;
    Debug.Log($"Condition={condition}, Time_s={elapsed:F3}");
}
```

`BeginTrial()` wywołuj przy pokazaniu celu, a `ConfirmTrial()` po jego potwierdzeniu. Dodatkowo zapisuj nietrafione wybory, anulowania i kolejność warunków. W kolejnych osobach równoważ kolejność A/B, aby ograniczyć efekt uczenia się.

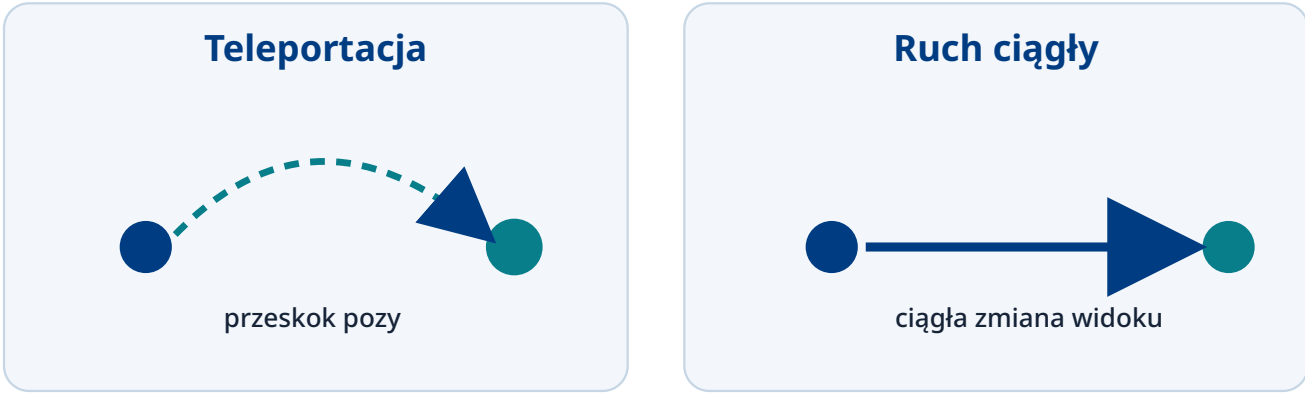
Jak oceniać interakcję: skuteczność, czas i koszt błędu



Protokół A/B: te same cele + równoważona kolejność

- **Skuteczność:** odsetek poprawnie zakończonych prób
- **Czas:** od wskazania celu do potwierdzonego wykonania; jawnie opisz początek i koniec
- **Błąd:** przypadkowe aktywacje, korekty, anulowania i konsekwencje pomyłek

Lokomocja i konflikt sensoryczny



Ruch ciągły: może wywoływać konflikt bodźców wzrokowych i przedsionkowych
Teleportacja: ogranicza ciągły ruch obrazu, ale wymaga podtrzymania orientacji w przestrzeni

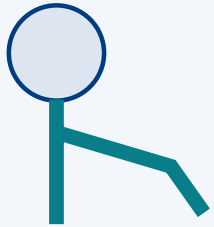
HMI przestrzenne



Projektując panel przestrzenny, kontroluj **odległość, kontrast, okluzję** i widoczność informacji krytycznych, nie tylko współrzędne obiektu

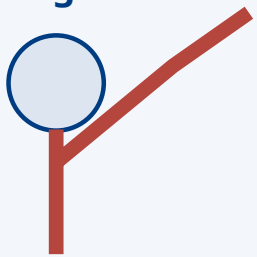
Ergonomia: gorilla arm

W zasięgu komfortu



ramię niżej

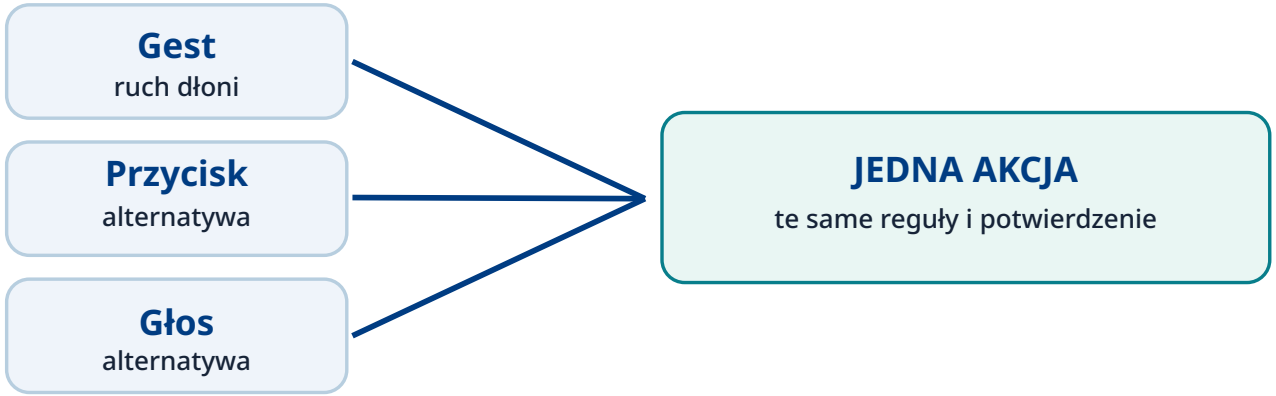
Długo uniesiona ręka



narastające zmęczenie

Długie unoszenie ramion może powodować zmęczenie (*gorilla arm*). Powtarzalne czynności projektuj w komfortowym zakresie ruchu.

Dostępność poprzez multimodalność



Zapewnij alternatywne sposoby wykonania czynności, jeżeli modalność wymaga sprawności niedostępnej części użytkowników

Dodatkowy kanał musi prowadzić do tej samej czynności



Alternatywa dla gestu może wykorzystywać przycisk, klawiaturę lub przełącznik

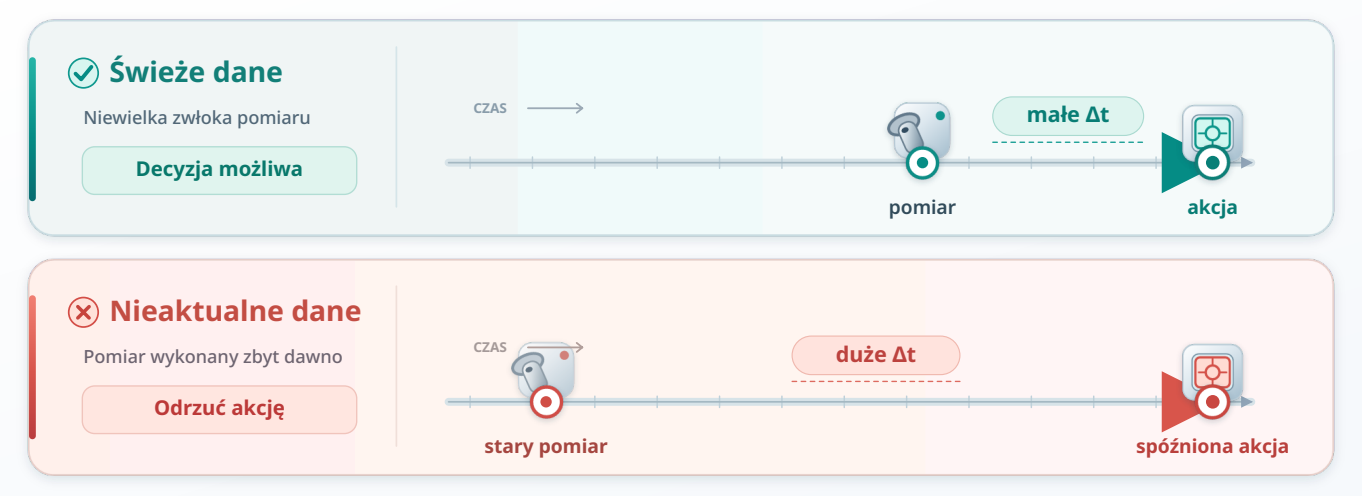
Ten sam cel, anulowanie i potwierdzenie muszą pozostać dostępne. Zwiększenie liczby modalności bez sprawdzenia ich zgodności może zwiększać liczbę pomyłek.

Spojrzenie a prywatność



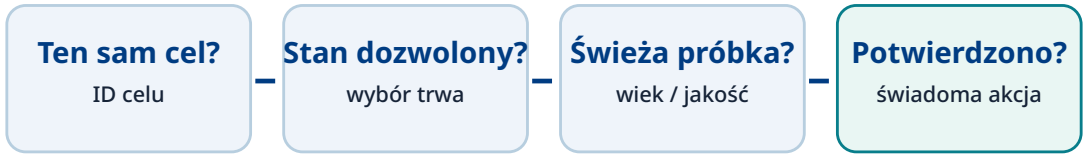
Dane eye trackingu mogą ujawniać zachowania użytkownika (fenotypowanie cyfrowe). Zbieraj wyłącznie dane niezbędne do realizacji zadania

Świeżość danych interakcji



Poprawnie rozpoznany gest może skutkować niebezpieczną komendą przy nieaktualnej pozie człowieka lub robota. Kontroluj czas pomiaru, transformacje i stan komunikacji.

Bramka wykonania: czy wolno teraz uruchomić akcję?



Dowolne NIE → anulowanie; tylko 4 × TAK → wykonanie

Warunki **muszą być spełnione równocześnie**: ten sam cel, dozwolony stan, świeża próbka i odrębna aktywacja. Demonstrator zakłada **jeden interaktor i jeden obiekt**; nie jest bramką bezpieczeństwa robota.

W demonstratorze: `hovered && selected && !consumed && IsFresh()`.

Jeżeli warunek zawiedzie, **odrzuć aktywację**. Po przyjęciu zablokuj ponowne wykonanie aż do nowego wyboru.

Podsumowanie

Modalność interakcji dobieramy do zadania i niepewności sygnału.

Komfort, dostępność i prywatność są wymaganiami projektowymi.

Wiarygodność polecenia wymaga świeżego stanu świata.

Studium przypadku: aktywacja po utracie śledzenia

1. `HoverEnter` i `SelectEnter`: użytkownik wskazał i wybrał cel.
2. `ReportTrackingLost()`: ostatnia poza traci ważność, wybór zostaje odrzucony.
3. Spóźnione `Activate()`: **odmowa wykonania**, bez wywołania `onAccepted`.
4. Dopiero nowa, świeża próbka i **nowy wybór** pozwalają ponowić aktywację.

Demonstrujemy ochronę logiki interfejsu, nie funkcję bezpieczeństwa robota.

Bibliografia

- Unity, XR Interaction Toolkit.
- MacKenzie (1992), *Fitts' Law as a Research and Design Tool in HCI*, doi:10.1207/s15327051hci0701_3.
- LaValle, Virtual Reality: interakcja.