

# Zastosowanie technologii Leap Motion w grze komputerowej

## Wnioski końcowe

### 1. **Wnioski na podstawie procesu tworzenia gry komputerowej przy pomocy technologii Leap Motion**

Podczas procesu tworzenia interaktywnej gry komputerowej z zastosowaniem kontrolera Leap Motion, mogliśmy prześledzić różne czynniki wpływające na potencjalny sukces tej technologii w opisywanym zagadnieniu. Projekt został napisany w języku C#, jako że był to jeden z języków, w którym dostępne jest Leap Motion SDK. API dostarczone wraz z kontrolerem jest przyjazne i proste w obsłudze. Dlatego też pisanie samej aplikacji było doświadczeniem ciekawym i raczej mało czasochłonnym.

Problematyczny okazał się fakt, że listener Leap Motion działa w osobnym wątku, a także reakcje na konkretne zdarzenia i gesty również możemy przechwycić jedynie w tamtym wątku. Sprawiało to poważny problem w C#, gdyż jak wiadomo, wszelkie operacje na graficznym interfejsie muszą być wykonywane w wątku głównym (UI-thread). Ponieważ konkretne ruchy wykrywane przez Leap Motion działały na model programu w pierwszej kolejności, a sam widok był bindowany do tegoż modelu, przechodzenie między wątkami w celu wyświetlania zmian w interfejsie graficznym okazało się dość skomplikowane. Nie jest to sam w sobie problem Leap Motion, ale połączenia konkretnego języka programowania i tej technologii.

### 2. **Testy praktyczne kontrolera Leap Motion**

Ponieważ kluczowym elementem naszej interaktywnej gry miało być wykrywanie predefiniowanych gestów, a konkretniej jednego ruchu, tzw. *swipe*, bardzo duża odpowiedzialność spoczywała na samym kontrolerze, by wykonywane przez gracza ruchy zostały odczytane należycie i z odpowiednią dużą szybkością.

Niestety, po zintegrowaniu gry z kontrolerem i uruchomieniu aplikacji okazało się, że skuteczność wykrywanych gestów nie jest idealna. Ich wykrywalność była kluczowa dla naszej aplikacji, dlatego postanowiliśmy zbadać problem ich wykrywania.

W tym celu wykonywaliśmy określone gesty przed kontrolerem i sprawdzaliśmy skuteczność ich wykrycia przez urządzenie. Za

sensowną minimalną wartość przed rozpoczęciem doświadczenia przyjęliśmy 90%. Jakikolwiek wynik poniżej 60% uznaliśmy za wartość krytyczną uniemożliwiającą sensowną rozgrywkę. Wyniki wyrażone w tabeli poniżej to procent ruchów poprawnie zinterpretowanych przez kontroler.

	Ruch <i>swipe</i>	Ruch <i>circle</i>	Ruch <i>screen tap</i>	Ruch <i>key tap</i>
Jedna ręka. Mała szybkość.	85	60	92	87
Dwie ręce. Mała szybkość.	65	60	80	83
Jedna ręka. Średnia szybkość.	91	77	84	87
Dwie ręce. Średnia szybkość.	76	71	82	83
Jedna ręka. Duża szybkość.	32	46	57	40
Dwie ręce. Duża szybkość.	25	39	52	54

### 3. Podsumowanie zalet i wad użycia Leap Motion w grach

Do niewątpliwych zalet kontrolera należy przede wszystkim intuicyjność sterowania dłonią. Precyzja urządzenia jeśli chodzi o pozycję dłoni jest duża, a informacje interpretowane są błyskawicznie. Interfejs API nie jest trudny do zrozumienia i jest dostępny w wielu językach programowania.

Niestety znacznie więcej jest wad. Najważniejsza z nich wynika z fatalnego wykrywania predefiniowanych gestów. Samych gestów jest jedynie 4, a tworzenie własnych wymaga sporych nakładów pracy. W dodatku gest *swipe* jest bardzo podobny do *circle*, co powoduje że żadna gra nie może się opierać na rozróżnianiu ich, gdyż wielokrotnie są wykrywane jednocześnie.

Kolejnym problem z wykrywaniem gestów jest fakt, że jednorazowy pojedynczy ruch zostaje wykryty jako wiele różnych gestów tego samego typu. Tym samym odpowiedzialność za wzięcie pod uwagę tylko jednego (faktycznego) gestu jest przerzucona na dewelopera. Konieczne jest badanie opóźnień, kierunków ruchu etc., czyli czynności, które powinny zostać zaimplementowane w samym kontrolerze.

Innym problemem jest bardzo wąski zakres wykrywania ręki. Poruszając dłonią i kontrolując jej pozycję łatwo jest zapomnieć, że zakres widoczności przez kontroler nie jest nieograniczony i sprowadza się do około 30 cm od urządzenia. Projektując grę trzeba mieć to na uwadze, by nie powodować sytuacji w których zgubiona przez kontroler dłoń w sposób nagły i drastyczny wpływa na rozgrywkę.

Jeśli chodzi o wszelkiego rodzaju zakłócenia, na przykład pojawienie się głowy nad kontrolerem równocześnie z dłonią, albo zbyt jasnym oświetleniem, wszystkie te czynniki bardzo wyraźnie wpływają na pracę Leap Motion. Urządzenie potrafi wykryć dłoń tam, gdzie jej nie ma (np. uznać za dłoń wspomnianą głowę czy kwiatek!), często też podczas prób zawieszało się i konieczne było ponowne uruchamianie. Wszystkie powyższe sytuacje są absolutnie niedopuszczalne podczas gry.

Ostatecznym problemem tej technologii jest zmęczenie użytkownika: niezależnie od sposobu wykorzystanie tej technologii, gracz bardzo szybko zmęczony się ciągłym trzymaniem dłoni w górze.

#### **4. Możliwe zastosowania technologii Leap Motion przy projektowaniu systemów komputerowych**

Wcześniej nie został wspomniany problem wykorzystania technologii Leap Motion w jakiegokolwiek rzeczywistej branży. Trzeba przyznać, że pomysł sensownego wykorzystania kontrolera nie jest taki trywialny. Z uwagi na fakt, że wykrycie dłoni jest procedurą ciągłą, i jeśli nasza dłoń wykona ruch w prawo, to nie jest możliwe, by ten ruch ponowiła nie wykonując wpierw ruchu w lewo, Leap Motion wydaje się nadawać do systemów, gdzie nasza dłoń pełni funkcję symulatora i sterownika, np. jakiegoś urządzenia. Pojawia się tylko pytanie, w czym jest to lepsze niż standardowy joystick? Otóż, niestety, jest nawet gorsze. Gorszy poziom kontroli, większa możliwość wykonania przypadkowego ruchu, niski poziom bezpieczeństwa i znacznie większe ryzyko złej interpretacji ruchu dłoni.

Co więc zostaje? Może przeglądanie dużych zbiorów danych przy pomocy dłoni? Albo wybór palcem konkretnego zbioru na ekranie? Od tego jest jednak mysz komputerowa, w dodatku ręka się nie męczy bo spoczywa NA urządzeniu.

#### **5. Końcowe podsumowanie**

Nie można nie odnieść wrażenia że ta technologia nie ma prawie żadnego zastosowania. Mogłoby nim być coś wykorzystującego koordynację i synchronizację ruchów wszystkich palców jednocześnie, albo użycia dwóch dłoni naraz (czynność byłaby synchronizowana, np.

jedną ręką coś przytrzymuje podczas gdy druga zaznacza). Mysz komputerowa nie daje nam możliwości zaznaczania więcej niż jednego punktu na raz. Ale wszystkie opisane wyżej wady i BARDZO słaba zawartość własnej kontrolery i dostępnych predefiniowanych przez autorów możliwości, wysoka cena i właściwie zerowe obecnie uznanie na świecie powodują, że tak jak w popularnej opinii w sieci możemy jedynie potwierdzić, że Leap Motion JEST obecnie rynkową porażką.