



---

Przedmiot: Interfejsy człowiek-komputer

## HCI2025\_1\_GAME

Temat projektu: Realizacja gry „CyberBug Combat” inspirowanej mechaniką „Vampire Survivors”

Spis treści:

1.	ABSTRAKT .....	2
2.	WSTĘP .....	3
3.	REALIZACJA PROPONOWANEGO ROZWIĄZANIA I TESTY .....	4
4.	OPIS MECHANIKI GRY .....	6
5.	PODSUMOWANIE I WNIOSKI .....	10
6.	DODATEK A: PROJEKT GRY .....	10

Wykonali: Adam Jędrzejczyk, Patryk Kamiński, Iwo Rymer, Mikołaj Pietrzyk

4 rok *AiR*

konsultant: *dr inż. Jaromir Przybyło*

Wersja 1.0

Kraków, maj 2025.

## 1. Abstrakt

Celem projektu było stworzenie gry komputerowej inspirowanej mechaniką popularnej produkcji „Vampire Survivors”, wzbogaconej o własne modyfikacje rozgrywki oraz unikalny system sterowania. Kluczowym aspektem przedsięwzięcia była implementacja komunikacji z niezależnymi grupami projektowymi, które opracowały alternatywne interfejsy sterowania grą za pomocą gestów palcem, ruchów głowy oraz komend głosowych.

Gra została napisana w silniku Godot i w pełni obsługiwała odbiór sygnałów z tych systemów, prawidłowo je interpretując i odzwierciedlając w działaniach w grze. Koordynacja pomiędzy grupami przebiegła sprawnie, co umożliwiło płynną integrację i testowanie różnych metod interakcji. Najważniejszym rezultatem projektu było potwierdzenie, że skuteczne sterowanie grą może odbywać się za pomocą wielu intuicyjnych metod, nie ograniczając się wyłącznie do tradycyjnych urządzeń wejściowych.

Wnioski płynące z realizacji wskazują, że techniki takie jak rozpoznawanie gestów, ruchów ciała czy komend głosowych mogą być z powodzeniem stosowane w gotowych aplikacjach, oferując użytkownikom bardziej dostępne i elastyczne sposoby interakcji z komputerem.



**Rys. 1.** Zrzut ekranu z rozgrywki. Postać gracza (Techno-mag na środku) jest atakowany przez przeciwników.

## 2. Wstęp

Celem projektu było stworzenie gry komputerowej umożliwiającej implementację różnych form alternatywnego sterowania. Ogólną koncepcję gry wyłoniono w drodze głosowania spośród 13 pomysłów opracowanych podczas wspólnej „burzy mózgów”, w której uczestniczyły wszystkie zespoły projektowe. Zdecydowano, że będzie ona oparta na walce z przeciwnikami w środowisku 2D. Jej tytuł miał nosić nazwę „CyberBug Combat”.

Akcje defensywne i ofensywne głównej postaci miały być wyzwalane na podstawie różnych modalności, takich jak gesty, pozycje ciała czy mimika twarzy (dla przykładu uśmiech = tarcza, uniesione ręce = atak, gest serca = leczenie), a także komendy głosowe, wykorzystywane do rzucania zaklęć lub uruchamiania specjalnych umiejętności. Gracz miał dynamicznie reagować na pojawiających się przeciwników, dobierając odpowiednie formy interakcji. W zależności od sytuacji miał zmieniać wyraz twarzy, wykonać określony gest lub wypowiedzieć komendę głosową.

Inspiracją dla mechaniki rozgrywki stała się popularna gra z gatunku roguelike — „Vampire Survivors”. W ramach współpracy zespołowej poszczególne grupy podzieliły się odpowiedzialnością za realizację konkretnych modalności: sterowania głosem, gestem i ruchem ciała. Ustalono, że komunikacja pomiędzy komponentami będzie odbywać się za pomocą protokołu UDP z wykorzystaniem gniazd (socketów).

Dodatkowym etapem projektowania było określenie stylu graficznego oraz wybór silnika gry. Przyjęto stylistykę nawiązującą do motywów science fiction i cyberpunk, z naciskiem na estetykę małych robotów, cybernetyki i informatyki. Grafiki miały być generowane przy użyciu ogólnodostępnych modeli sztucznej inteligencji. Ze względu na prostotę projektu oraz jego eksperymentalny charakter, zdecydowano się na użycie silnika Godot. Nie wymaga on dużo zasobów i jest intuicyjnym narzędziem dla początkujących twórców gier.

### 3. Realizacja proponowanego rozwiązania i testy

Projekt był realizowany etapami, zgodnie z harmonogramem opartym na cotygodniowych konsultacjach, podczas których omawiano postępy oraz planowano kolejne działania. W początkowej fazie każdy członek zespołu zapoznał się z wybranym środowiskiem programistycznym - silnikiem Godot. Po wstępnym przygotowaniu, zespół podzielił się zadaniami według kompetencji i obszarów odpowiedzialności.

Jedna osoba została odpowiedzialna za przygotowanie wszystkich grafik, w tym postaci, elementów świata oraz efektów ataków. Kolejny członek zespołu zajął się implementacją komunikacji opartej na protokole UDP, a także współpracą z innymi grupami projektowymi w celu ustalenia wspólnego standardu pakietów danych, które miały być przesyłane pomiędzy modułami. Pozostali członkowie zespołu pracowali nad mechaniką gry: sposobem ataku i obrony głównej postaci, systemem pojawiania się przeciwników, ich zachowaniem, a także logiką zadawania i odbierania obrażeń.

W trakcie realizacji powstał standard komunikacji, obejmujący listę możliwych akcji i odpowiadających im komunikatów przesyłanych przez protokół UDP. Komunikaty były kodowane jako łańcuchy znaków ASCII w formacie (ID)(Data), gdzie ID oznaczało identyfikator grupy (0 - OpenFace, 1 - OpenPose, 2 - Voice), a Data zawierało treść komunikatu. Przykład takiej wiadomości: 01abs19348 (ID – 0, Data – 1abs19348).

Tabela 1. Specyfikacja potrzebnych sygnałów sterowania dla pozostałych zespołów projektowych, podane zostały wartości istotnych elementów wiadomości wysyłanych przez każdą z grup

Sterowanie	UDP message	Grupa, modalność
Ruch w prawo	„1” pod indeksem 3	Open Pose
Ruch w lewo	„1” pod indeksem 2	Open Pose
Ruch w górę	„1” pod indeksem 0	Open Pose
Ruch w dół	„1” pod indeksem 1	Open Pose
Akcja specjalna - leczenie	„4” pod indeksem 0	Voice
Akcja specjalna - tarcza	„3” pod indeksem 0	Voice
Akcja specjalna – super- atak	„1” pod indeksem 2	Open Face
Super-atak celowanie -	„1” pod indeksem 6	Open Face

prawo		
Super-atak celowanie - lewo	„1” pod indeksem 5	Open Face
Super-atak celowanie - górze	„1” pod indeksem 7	Open Face
Super-atak celowanie - dół	„1” pod indeksem 8	Open Face

Ustalono, że każda z grup projektowych będzie korzystać z własnego zakresu adresów IP oraz dynamicznych portów UDP, a adres serwera (komputera, na którym uruchomiona została gra) podany został przed rozpoczęciem testów. Przyjęte rozwiązanie problematyki komunikacji zakłada identyfikowanie poszczególnych klientów poprzez pierwszy znak wiadomości co sprawia, że znajomość adresu IP każdej z grup nie jest konieczna.

Mechanika gry była rozwijana równolegle z implementacją interfejsu graficznego i komunikacji sieciowej. Na początku umożliwiono podstawowe poruszanie się postaci w otwartym świecie. Sterowaniem postacią zajęła się grupa OpenPose. Następnie dodano system losowego pojawiania się przeciwników o różnych właściwościach, takich jak: punkty życia, typ ataku, szybkość poruszania się czy częstotliwość pojawiania się. Dodatkowo zaimplementowano dwa typy ataków gracza: podstawowy, który był wykonywany automatycznie i nie wymagał ingerencji, oraz „superatak”, którego kierunek należało określić. Za jego sterowanie odpowiedzialna została grupa OpenFace.

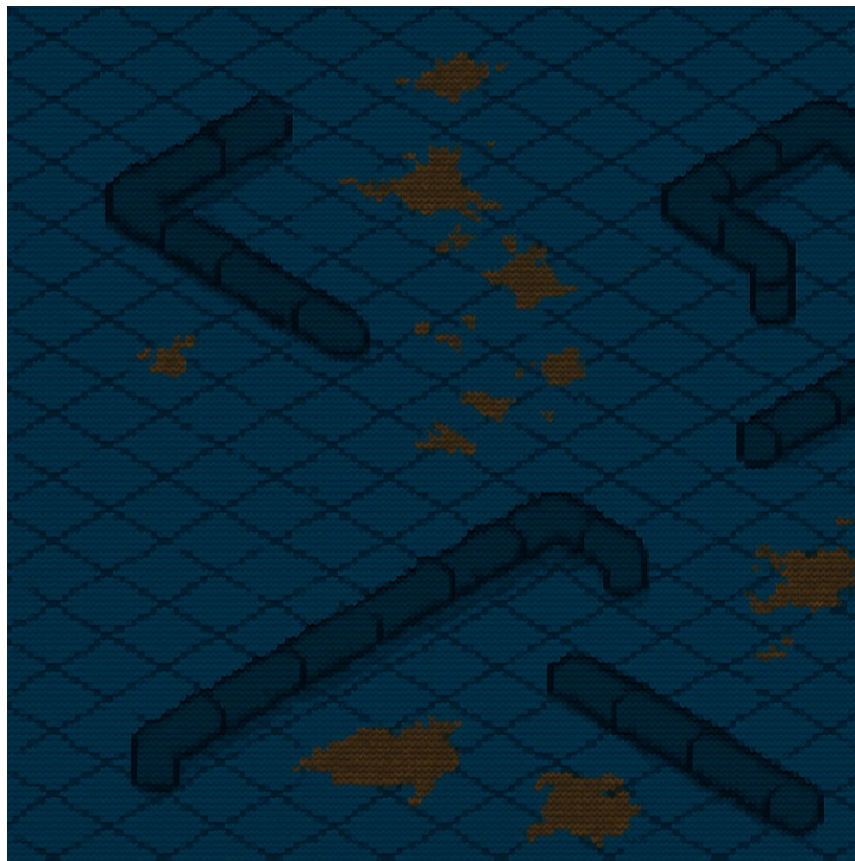
W późniejszych etapach wprowadzono także mechanikę leczenia (regeneracja zdrowia co określony czas) oraz tarczy (czasowa niewrażliwość na obrażenia), które mogły być aktywowane za pomocą komend głosowych. Każda z tych funkcjonalności była również dostępna w trybie testowym poprzez konwencjonalne sterowanie z klawiatury.

Testy końcowe przeprowadzono w sali laboratoryjnych z udziałem wszystkich grup projektowych, korzystających ze wspólnej sieci lokalnej (routera). Po wprowadzeniu drobnych poprawek wynikających z niejednoznacznej interpretacji formatu danych, system komunikacji został w pełni zsynchronizowany, a projekt funkcjonował zgodnie z założeniami. Zrealizowana gra prawidłowo odbierała sygnały z różnych źródeł (gesty, głos, mimika), reagując na nie w czasie rzeczywistym i umożliwiając płynną rozgrywkę.

## 4. Opis mechaniki gry

### Świat gry

Rozgrywka w grze zachodzi na mapie 2D widzianej z izometrycznie lotu ptaka. Postać gracza porusza się po tej mapie, walcząc z przeciwnikami którzy stale pojawiają się (spawnują się) w losowym miejscu na mapie poza zasięgiem widzenia gracza.



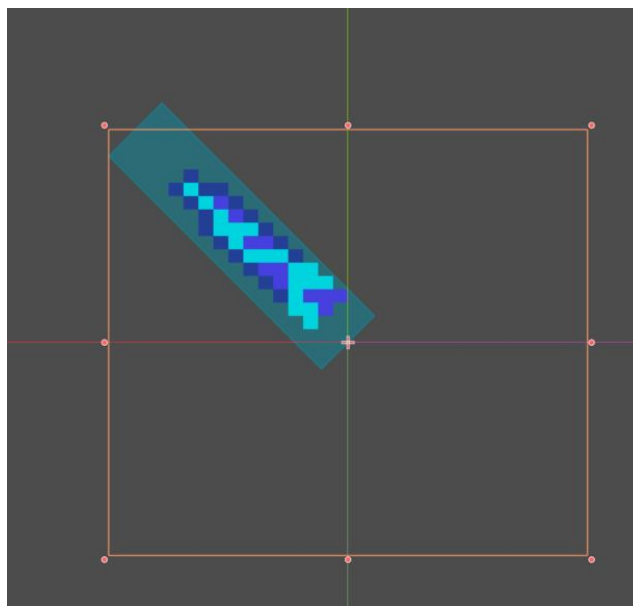
**Rys. 2.** Obraz wykorzystany jako tło rozgrywki.

### Ruch postaci

Postać porusza się w jednym z 8 dostępnych kierunków (góra/dół/prawo/lewo) oraz na ukos, gdy gra uzyskuje sygnał sterujący odpowiadający za ruch w dwóch sąsiednich kierunkach (na przykład jednocześnie przesyłany jest sygnał „w górę” i „w lewo”).

### Atak podstawowy

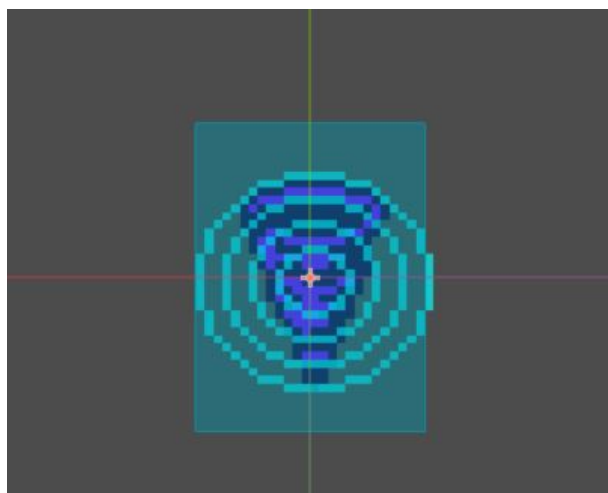
Podstawowym atakiem postaci jest lanca plazmowa, którą postać strzela automatycznie w losowego przeciwnika w zasięgu wykrycia. Atak zadaje małe obrażenia w porównaniu z innymi atakami, ale nie wymaga ingerencji gracza.



**Rys. 3.** Widok 2D sceny ataku podstawowego. Prostokątne obramowanie to obszar kolizji pocisku z celem.

### **Atak podstawowy – tornado EMP**

Postać posiada drugi atak podstawowy – jest nim tornado. Tornado cyklicznie generowane jest przez postać gracza (technomaga) i porusza się w kierunku w którym poruszała się postać tuż przed wysłaniem ataku. Ruch tornada jest zaburzany losową zmienną przez co trasa którą podąży ma element niepewności.

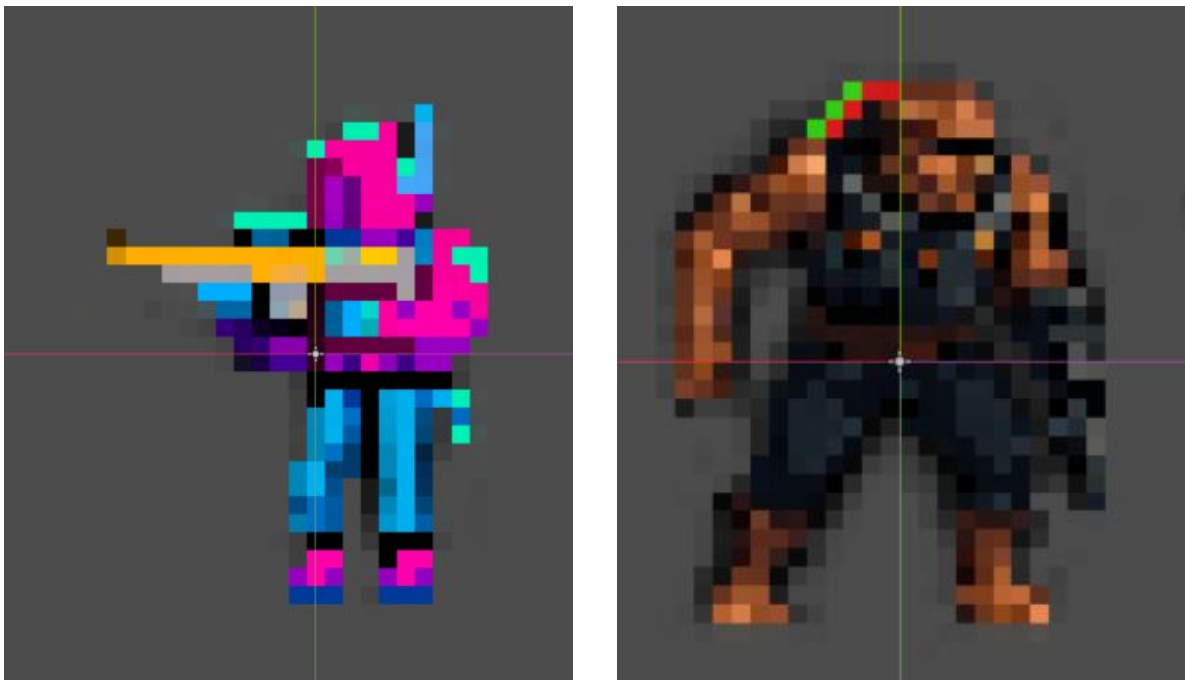


**Rys. 4.** Widok 2D sceny ataku tornada.

### **Przeciwnicy**

W grze występuje kilka typów przeciwników, których można podzielić na dwie kategorii: wrogowie walczący wręcz i wróg zasięgowy. Wrogowie walczący wręcz idą w kierunku postaci

gracza i zadają jej obrażenia poprzez kontakt bezpośredni, podczas gdy wróg zasięgowy zbliża się na zdefiniowany zasięg, po czym zatrzymuje się. Przeciwnicy mają określone poziomy życia i gdy stracą wszystkie punkty życia są usuwani z mapy.



**Rys. 5.** Przeciwnik zasięgowy (a), Przeciwnik walczący wręcz (mutant) (b)

### **Atak specjalny – laser**

Ostatnim z ataków jest laser – jest to atak specjalny którego kierunkiem można sterować, analogicznie do ruchu postaci (8 kierunków strzału). Laser zadaje duże obrażenia, ale ma określony zasób energii. Laser zużywa energię gdy strzela, i regeneruje zasób kiedy jest nieaktywny.

### **Leczenie**

Poza atakiem specjalnym, gracz ma do dyspozycji dwie akcje specjalne – leczenie i tarcza. Leczenie przywraca niewielką ilość życia, po czym jest niedostępne przez dłuższy czas.





**Rys. 6.** Moment strzału we wrogiego strzelca przy użyciu lasera.

### **Tarcza**

Tarcza sprawia że postać jest niewrażliwa na obrażenia.



**Rys. 7.** Postać po aktywacji leczenia z włączoną tarczą.

### **Śmierć**

Celem gracza jest przeżycie, jeśli jednak jego pula życia spadnie do zera, wyświetlana jest scena końca gry:



**Rys. 8.** Ekran końca gry (śmierci)

## 5. Podsumowanie i wnioski

Zrealizowany projekt w pełni spełnił swoje główne założenia. Gra działała poprawnie, zapewniając skuteczną integrację z rozwiązaniami opracowanymi przez pozostałe grupy projektowe. Dzięki wdrożeniu komunikacji opartej na protokole UDP możliwa była wymiana danych w czasie rzeczywistym pomiędzy grą, a zewnętrznymi systemami sterowania wykorzystującymi gesty, mimikę twarzy i komendy głosowe.

Choć sama gra mogłaby zostać rozbudowana o dodatkowe elementy, takie jak menu główne, system dźwięków czy bardziej zaawansowaną oprawę wizualną, nie były one celem nadrzędnym projektu. Priorytetem było stworzenie funkcjonalnej i elastycznej platformy testowej, umożliwiającej weryfikację niestandardowych metod sterowania opracowanych przez inne zespoły. W tym aspekcie projekt zakończył się sukcesem.

Wnioski płynące z realizacji wskazują jednoznacznie, że alternatywne metody interakcji z komputerem (takie jak rozpoznawanie gestów, mimiki czy mowy) mogą być z powodzeniem stosowane w rzeczywistych aplikacjach. Stanowią one ciekawe uzupełnienie lub nawet zamiennik tradycyjnych urządzeń wejściowych, oferując użytkownikom nowe, często bardziej dostępne i naturalne formy komunikacji z systemem.

## 6. DODATEK A: Projekt gry

Całość projektu jest dostępna pod linkiem:

[https://github.com/IwoRymer/ICK\\_2025\\_gra/tree/enemy\\_rev\\_plus](https://github.com/IwoRymer/ICK_2025_gra/tree/enemy_rev_plus)

Aby uruchomić projekt należy go pobrać z powyższego linku, przy czym należy wybrać gałąź **enemy\_rev\_plus**. Grę opracowano na wersji silnika Godot 4.4.1 na systemie operacyjnym Windows. Pobrany kod należy rozpakować a następnie zaimportować w interfejsie graficznym silnika.

Krótki film przedstawiający rozgrywkę dostępny jest pod linkiem:

<https://drive.google.com/file/d/1OIsdZ2n2-dTck1X8egb2NktO05K5ce0T/view?usp=sharing>