



Air Hockey

Grzegorz Kokoszka

Jakub Mendel

Piotr Okoń

Kamil Wojciechowski

Podział prac

- Jakub Mendel – szkielet aplikacji, UI, przetwarzanie obrazu
- Piotr Okoń – wygląd i logika gry
- Grzegorz Kokoszka, Kamil Wojciechowski - tryb multiplayer

Założenia projektu

- Możliwość gry z AI lub innymi graczami.
- Realistyczność fizyki odbicia piłki przez paletki.
- Czat w lobby przed włączeniem gry w trybie multiplayer.
- Płynny przebieg gry podczas gry sieciowej.

Sposób rozwiązywania problemów.

Sposób przetwarzania obrazu z kamery z użyciem biblioteki OpenCV tak, aby wskaźniki umożliwiały sterownie paletkami.

Zapamiętanie poprzedniej i odczyt nowej ramki z kamery

Odbicie lustrzane ramki poziomo

Utworzenie ramki **diff** = $|\text{actual} - \text{last}|$

Erozja ramki **diff**

Progowanie ramki **diff** na podstawie zadanej wartości

Konwersja ramki **diff** do skali szarości

Ustawienie na ramce **diff** białych pikseli w pozycjach aktualnych punktów wskaźnika

Rozszerzenie ramki **diff**

Konwersja ramki **actual** do przestrzeni **HSV**

Sposób rozwiązania problemów.

Utworzenie kopii
ramki **actual** do
skali szarości

Wyszukanie
okręgów i
utworzenie mapy
bitowej **circleMask**

Utworzenie mapy
bitowej **dst** metodą
cv::inRange(...) na
podstawie obrazu **HSV**

Pomnożenie mapy
dst przez **diff**

Dodanie bitowe
mapy **circleMask**
do **diff**

Filtracja mapy **dst**
przez erozję a
następnie
rozszerzanie

Wykrycie punktów
przez
cv::SimpleBlobDetector
na mapie **dst**

Filtracja
nadmiarowych
mniej znaczących
punktów

Przypisanie pozycji
punktów do
wskaźnika

Sposób rozwiązania problemów.

- Broadcast UDP na porcie 50001 dla danej sieci, pod jaką jesteśmy podłączeni (na przykład hamachi).
- Podczas tworzenia połączenia jako serwer możliwość wyboru interfejsu sieciowego , na którym będziemy broadcastować pakiety.
- Możliwość podłączenia się pod wybrany serwer wybierany z listy wszystkich możliwych serwerów.
- Socket TCP nasłuchuje na wszystkie połączenia pod adresem ip 0.0.0.0 (wychwytuje jakiekolwiek połączenie kierowane w jego stronę), w tym samym czasie gdy aktywny jest UDP broadcast, po wychwyceniu jakiegoś adresu ip z odpowiednim pakietem zrywane jest połączenie UDP i następuje komunikacja TCP.
- Po wybraniu odpowiedniego serwera przez klienta następuje „handshake” oraz przejście do lobby (zmiana komunikacji UDP na komunikację TCP).
- W lobby możliwość edycji gry Real-Time (update pól widocznych dla klienta po zmianie ustawień przez tworzącego grę takich jak ilość bramek potrzebnych do wygrania oraz rodzaj mapy).

Sposób rozwiązania problemów.

- Format wysyłanych pakietów: Enum (rodzaj następnych wiadomości) oraz pakiet sfml, na przykład:
 - Server wysyła następujący pakiet podczas gry:
 - `Sf::packet p << Game << sf::vector2f(dot1.x, dot1.y) << sf::vector2f(dot2.x, dot2.y) << sf::vector2f(ball.velocity, ball.angle);`
 - Server odbiera następujący pakiet podczas gry:
 - `Sf::packet p << Game << sf::vector2f(dot1.x,dot1.y) << sf::vector2f(dot2.x,dot2.y)`
 - Klient wysyła następujący pakiet podczas gry:
 - `Sf::packet p << Game << sf::vector2f(dot1.x,dot1.y) << sf::vector2f(dot2.x,dot2.y)`
 - Klient odbiera następujący pakiet podczas gry:
 - `Sf::packet p << Game << sf::vector2f(dot1.x,dot1.y) << sf::vector2f(dot2.x,dot2.y) << sf::vector2f(ball.velocity, ball.angle)`
- Podczas wyłączenia serwera lista dostępnych serwerów automatycznie się odświeża (zostaje wysłany pakiet zawierający informacje o końcu broadcasting'u).
- Komunikacja UDP, jak i TCP jest organizowana na gniazdach nieblokujących.
- Symulacja gry przeprowadzana jest po stronie serwera, natomiast po stronie klienta wykonywane są obliczenia takie jak interpolacja i movement prediction, które mają na celu upłynnić grę.
- Po zakończeniu gry możliwość kontynuacji (powrót do lobby) lub przejścia z powrotem w tryb broadcasting'u UDP.

Harmonogram prac

- 06.03 – 19.03 Zapoznanie się z tematem
- 20.03 – 31.03 Prototyp wykrywania punktów
- 01.04 – 07.04 Prototyp Air Hockeya
- 01.04 – 14.04 Tworzenie projektu z komunikacją sieciową
- 07.04 – 14.04 Rozbudowa metody wykrywania punktów, GUI
- 14.04 – 21.04 Rozbudowa gry, implementacja painta
- 21.04 – 01.05 Dostosowanie komunikacji sieciowej do rozbudowanej wersji gry
- 01.05 – 14.05 Scalenie projektów komunikacji z projektem Air Hockeya
- 14.05 – 04.06 Refaktoryzacja kodu, ostatnie poprawki

Koniec prezentacji.

Czas na pytania —)