

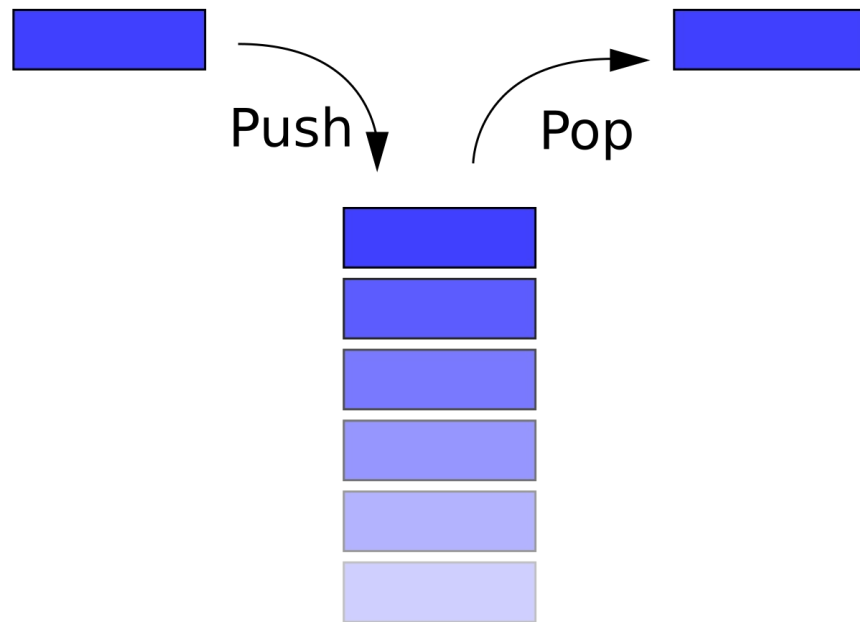
Stosy i Kolejki – Makao

Michał Błaszczewicz, Informatyka rok I, L01

Czym jest Stos?

Stos jest strukturą danych,
do której można dokładać
na konie elementy i z której
końca można odbierać
elementy.

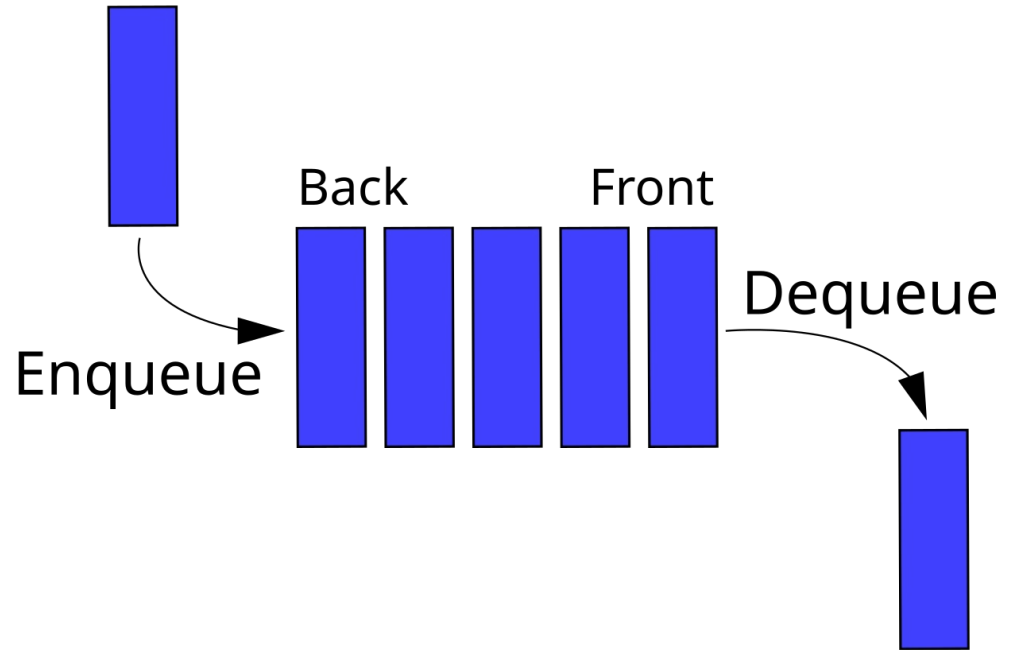
LIFO
(**L**ast **I**n **F**irst **O**ut)



Czym jest Kolejka?

Stos jest strukturą danych, do której można dokładać na koniec elementy i z której początku można odbierać elementy.

FIFO
(**F**irst **I**n **F**irst **O**ut)



Czemu Makao?

- Talia działa jak Stos
- Lubię gry karciane
- Lubię robić gry
- Nie będę się musiał kłócić o zasady gry

Jako implementację Stosu użyłem `std::vector`

Założenia projektu

- Gra powinna być grywalna w terminalu (TUI)
- Gra powinna mieć zmienną liczbę graczy i używanych talii
- Gra powinna mieć wszystkie podstawowe elementy Makao:
 - Atakowanie dobieraniem kart
 - Żądanie kart
 - Blokowanie kolejek
 - „Dama na wszystko, wszystko na Damę”
 - Joker

Kod

- Dwa główne obiekty:
 - `makao : : Game` odpowiadający za stan zmiennych
 - `makao : : GameState` odpowiadający za przebieg rozgrywki
- Dodatkowe Klasy
 - `makao : : Card` odpowiadający za wartości kart
 - `makao : : Player` odpowiadający za graczy

Inicjacja Gry

```
1 makao::Game::Game(int deck_count, int player_count)
2 {
3     // Create cards
4     this->GeneratePlayingDeck(deck_count);
5
6     // Shuffle deck
7     this->Shuffle();
8
9     // Create players
10    for (int i = 0; i < player_count; i++)
11    {
12        this->players.push_back(makao::Player());
13    }
14    this->players[0].is_computer = false;
15
16    //Deal cards to players
17    for(int i = 0; i < 5; i++){
18        for(int p = 0; p < this->players.size(); p++){
19            this->players[p].hand.push_back(this->deck.back());
20            this->deck.pop_back();
21        }
22    }
23
24    //Setting up initial variables:
25    this->turn_counter = 0;
26
27    return;
28 }
```

```
1 void makao::Game::GeneratePlayingDeck(int deck_count)
2 {
3     for (int i = 0; i < deck_count; i++)
4     {
5         for (int c = 0; c <= 3; c++)
6         {
7             for (int s = 1; s <= 13; s++)
8             {
9                 makao::Card temp_card = makao::Card(s, c);
10                this->all_cards.push_back(temp_card);
11            }
12        }
13    }
14
15    for (int i = 0; i < this->all_cards.size(); i++)
16    {
17        this->deck.push_back(&this->all_cards[i]);
18    }
19    return;
20 }
```

Tasowanie Kart

```
1 void makao::Game::Shuffle()
2 {
3     //Add discarded cards back to the deck
4     while(this->discarded.size() > 0){
5         this->deck.push_back(this->discarded.back());
6         this->discarded.pop_back();
7     }
8
9     // Implementation of Fisher-Yates algorithm :
10    // https://en.wikipedia.org/wiki/Fisher-Yates\_shuffle
11    std::srand(unsigned(std::time(NULL)));
12    int deck_size = this->deck.size();
13    for (int i = 0; i < deck_size - 1; i++)
14    {
15        int j = (rand() % (deck_size - i)) + i;
16        std::swap(this->deck[i], this->deck[j]);
17    }
18 }
19 }
20
```


Granie Kart

```
1 void makao::GameState::PlayCard(makao::Card *played_card)
2 {
3     // Reset Color Override
4     this->color_override.reset();
5
6     this->card_played_this_round = true;
7
8     // Evaluate card
9     switch (played_card->shape)
10    {
11        // Draw attacks
12        case makao::Card::SHAPE::TWO:
13            this->current_attack = makao::GameState::ATTACK_TYPE::DRAW;
14            this->attack_strength += 2;
15            break;
16        case makao::Card::SHAPE::THREE:
17            this->current_attack = makao::GameState::ATTACK_TYPE::DRAW;
18            this->attack_strength += 3;
19            break;
20        case makao::Card::SHAPE::KING:
21            this->current_attack = makao::GameState::ATTACK_TYPE::DRAW;
22            this->attack_strength += 5;
23            break;
24
25        // Halt attacks;
26        case makao::Card::SHAPE::FOUR:
27            this->current_attack = makao::GameState::ATTACK_TYPE::HALT;
28            this->attack_strength += 1;
29            break;
30
31        // Require attack
32        case makao::Card::SHAPE::JACK:
33            this->current_attack = makao::GameState::ATTACK_TYPE::REQUIRE;
34            this->required_card = this->GetRequestedShape(this->root_object->players[(this->root_object->turn_counter - 1) % this->root_object->players.size()].is_computer);
35            break;
36
37        // Change Color
38        case makao::Card::SHAPE::ACE:
39            this->current_attack = makao::GameState::ATTACK_TYPE::NONE;
40            this->color_override = this->GetRequestedColor(this->root_object->players[(this->root_object->turn_counter - 1) % this->root_object->players.size()].is_computer);
41            break;
42
43        // Other Cards
44        default:
45            this->attack_strength = 0;
46            break;
47    }
48
49    std::cout << "Player " << ((this->root_object->turn_counter - 1) % this->root_object->players.size()) + 1 << " played " << played_card->getIndicator() << std::endl;
50
51    this->root_object->discarded.push_back(this->current_card);
52    this->current_card = played_card;
```

AI przeciwników

```
1  if (this->root_object->players[player_id].is_computer == true)
2      {
3          // Computer's turn
4
5          makao::Player *player = &this->root_object->players[player_id];
6
7          std::vector<makao::Card *> accepted_hand;
8
9          do
10         {
11
12             accepted_hand = this->GetPlayableCards();
13             if (accepted_hand.size() > 0)
14             {
15                 int choice = (rand() % accepted_hand.size());
16
17                 this->PlayCard(accepted_hand[choice]);
18                 player->hand.erase(std::remove(player->hand.begin(), player->hand.end(), accepted_hand[choice]), player->hand.end());
19             }
20         } while (accepted_hand.size() > 0);
21
22         this->EndTurn();
23
24         return;
25     }
```

Wychodzenie z programu

```
1  if (player->hand.size() == 0)
2  {
3      std::cout << "Player " << ((this->root_object->turn_counter - 1) % this->root_object->players.size()) + 1 << " has won!" << std::endl;
4      throw makao::GameState::QuitGame();
5      // Yup, this is how the game ends. I really dislike the idea but my semi-structured code would be way too much of a mess if i were to exit it normally.
6  }
```

```
1  #include <iostream>
2  #include "include/game.hpp"
3
4  int main()
5  {
6
7      try
8      {
9          makao::Game GameObject = makao::Game();
10
11          makao::GameState GameStateObject = makao::GameState(&GameObject);
12
13          GameStateObject.InitiateGameLoop();
14      }
15      catch (makao::GameState::QuitGame &e)
16      {
17          return EXIT_SUCCESS;
18      }
19  }
20
```

Sukcesy

- Program działa i jest grywalny
- Większość funkcji została zaimplementowana
- Zaimplementowano zmianę kolorów (Asy)
- Program został napisany w czysty sposób i wydaje się być przystosowany do skalowania.

Problemy

- Nieintuicyjny sposób działania kompilatorów
- Brak odpowiednich integracji do VSCodium

ROZWIĄZANIE:

- Przerzucenie się na VSCode by korzystać z wtyczek Microsoftu
- Używanie CMake

Problemy

- Problemy z linkowaniem bibliotek
- Cykliczne Forward Declaration
- Wielokrotna redeklaracja obiektów i metod

ROZWIĄZANIE:

- Przepisanie projektu by korzystał z headerów i plików źródłowych
- Używanie CMake

Problemy

- Błędy niezrozumiałe:
 - Segmentation fault
 - Illegal instruction
 - Floating point precision error

ROZWIĄZANIE:

- Korzystanie z GDB
- Przepisanie kodu używającego `std::cin` na `std::getline`

Wyniesiona Nauka

- Korzystanie i lepsze zrozumienie wskaźników i referencji
- Nauka korzystania z CMake
- Nauka konceptu, tworzenia oraz importowania headerów i plików źródłowych
- Nauka korzystania z GDB

Wnioski

- Projekt był za duży jak na środowisko z którym nie byłem bardzo zaznajomiony
- C++ nie jest językiem przyjaznym zwykłemu użytkownikowi, ale jak się człowiek nie boi terminala to traci dużo swojej grozy
- Większość moich problemów w tym projekcie by nie istniała gdybym korzystał z języka interpretowanego

Źródła które pomogły mi w pisaniu kodu

- [GeeksforGeeks.org](https://www.geeksforgeeks.org/)
- [W3schools.com](https://www.w3schools.com/)
- [Cplusplus.com](https://www.cppreference.com/)
- [Cplusplus.com](https://www.cplusplus.com/)
- [Linuxquestions.org](https://linuxquestions.org/)
- Brave AI
- [Techiedelight.com](https://www.techiedelight.com/)
- [Cmake.org](https://cmake.org/)
- [Wikipedia.org](https://www.wikipedia.org/)
- [Stackoverflow.com](https://stackoverflow.com/)
- [Programiz.com](https://www.programiz.com/)
- [Sentry.io](https://sentry.io/)
- Inne strony, na które mogłem wejść poszukując wiedzy

Źródła prezentacji

- Wikipedia „Stos (informatyka)” – 15.05.2025
- Wikipedia „Kolejka (informatyka)” – 15.05.2025