

# Technologia Programowania 2016/2017 – Lista 4 (lab)

## Termin: przed 16 grudnia

„Ameryka gra w pokera, Europa w brydża, Rosja w szachy, a Chiny w go. Pytanie: kto wygra?”

Cytat, ale nie pamiętam źródła.

### Gra Go

Zaprojektuj i napisz system do przeprowadzania rozgrywek w [grę go](#). Podstawowe zasady gry i wymagania dotyczące systemu są przedstawione poniżej. Warto na wstępie przejść krótki interaktywny [kurs go](#).

Zadanie należy rozwiązać **w parach**<sup>1</sup> wykorzystując repozytorium. Historia zmian w repozytorium będzie podstawą do oceny wkładu członków zespołu. Jeśli nie masz partnera zgłoś się do prowadzącego laboratorium.

**Zasady gry** (w razie wątpliwości proszę spojrzeć na [oficjalne zasady japońskie z komentarzami](#))

1. W jednej grze uczestniczy dwóch graczy. Gra toczy się planszy (goban) będącej siatką 19 poziomych i 19 pionowych linii tworzących 361 przecięć. Czasami gra się także na planszach 13 na 13 lub 9 na 9.
2. Gracze kładą na przemian czarne i białe kamienie na przecięciu linii. Rozpoczynają czarne. Plansza jest początkowo pusta. Celem gry jest otoczenie własnymi kamieniami obszaru większego niż obszar przeciwnika.
3. Kamieni raz postawionych na planszy nie można zabrać ani przesunąć, mogą natomiast zostać **uduszone** przez przeciwnika, jeśli tracą wszystkie **oddechy**. Po zabraniu ostatniego oddechu przeciwnik zabiera z planszy i przechowuje uduszone przez siebie kamienie (zwane **jeńcami**). Oddechem kamienia nazywamy niezajęte sąsiednie przecięcie (połączone linią z kamieniem). Na przykład kamień stojący samotnie na środku planszy ma cztery oddechy, a w rogu planszy - dwa. Kamienie jednego koloru stojące obok siebie i połączone liniami tworzą łańcuch, który ma wspólne oddechy – można je zbić albo wszystkie razem albo żadnego (zobacz np. [tutaj](#)).
4. Gracz nie może pozbawić swojej grupy kamieni ostatniego oddechu, ani położyć kamienia w punkt bez oddechu. Wyjątkiem od reguły jest sytuacja, w której taki ruch dusi kamienie przeciwnika (zobacz np. [tutaj](#)).
5. Kształt, w którym gracz może naprzemiennie dusić kamień przeciwnika, zwany jest **ko**. W celu uniknięcia nieskończonych cykli gracz, którego kamień został uduszony w ko, nie może udusić kamienia przeciwnika w tym ko w następnym ruchu (zobacz np. [tutaj](#)).
6. O kamieniach mówimy, że są żywe, jeżeli nie mogą być zbite przez przeciwnika. O kamieniach, które nie są żywe, mówimy, że są martwe. Puste punkty otoczone żywymi kamieniami tylko jednego gracza zwane są punktami wewnętrznymi. Wszystkie inne puste przecięcia są nazywane punktami niczymi. Kamienie, które są żywe, ale stykają się z punktami niczymi, są w **seki**. Punkty wewnętrzne otoczone przez żywe kamienie nie będące w seki są zwane **terytorium** (zobacz np. lekcje 5-9 [tutaj](#)).
7. Gracz zawsze może zrezygnować z ruchu. Gdy obaj gracze bezpośrednio po sobie zrezygnują z ruchu, gra się zatrzymuje. Następnie obaj gracze uzgadniają, które grupy kamieni są żywe, a które martwe, a także jakie są terytoria. Jeżeli gracze nie mogą dojść do porozumienia i jeden z graczy żąda wznowienia zatrzymanej gry, jego przeciwnik nie może odmówić i ma prawo zagrać jako pierwszy.

---

<sup>1</sup>Parowanie powinno się odbywać w ramach grup laboratoryjnych jednego prowadzącego.

8. Po uzgodnieniu, że gra się skończyła, każdy gracz usuwa ze swojego terytorium wszystkie kamienie przeciwnika i dodaje je do swoich jeńców. Następnie jeńców ustawia się w terytorium przeciwnika, po czym podlicza się i porównuje punkty terytorium. Gracz z większym terytorium wygrywa.<sup>2</sup>
9. W dowolnym momencie gry gracz może ją zakończyć poprzez przyznanie się do przegranej.

### Wymagania funkcjonalne

1. System powinien działać w oparciu o architekturę klient-serwer. Serwer powinien mieć możliwość obsługi wielu toczących się równolegle partii.
2. Gracz za pomocą aplikacji klienckiej powinien móc połączyć się z serwerem, sparować z innym graczem i przeprowadzić rozgrywkę. Aplikacja powinna też przedstawiać wszystkie informacje o toczącej się grze.
3. System nie powinien dopuszczać niepoprawnych ruchów. W punkcie 7. powyższych zasad gry, gracze mogą sami oznaczać martwe grup i zdobyte terytoria (wybory wymagają akceptacji przeciwnika). Dodatkowo dobrze by było, gdyby system sugerował domyślny podział grup i terytoriów.
4. Do gry mogą zostać dołączeni gracze-boty działający na serwerze. Stwórz jedną przykładową implementację bota, która będzie wykonywała przynajmniej poprawne ruchy.

### Dodatkowe wymagania

1. Zarówno klient i serwer powinny być napisane w Java. Kod powinien być dobrze skomentowany.
2. Postaraj się pisać projekt starannie i w przemyślany sposób. Projekty chaotyczne i pisane na ostatnią chwilę będą nisko ocenione. Bierz również pod uwagę, że to co stworzysz będzie miało wpływ na kolejną listę.
3. Postaraj się wykorzystać poznane na wykładzie narzędzia i wzorce. Im więcej ich zastosujesz, tym wyżej będzie ocenione Twoje rozwiązanie. Ich użycie powinno być jednak zawsze uzasadnione.
4. Stosuj podejście iteracyjne i przyrostowe i przygotuj się na ewentualne zmiany wymagań! Na każdych zajęciach do ostatecznego oddania programu powinien być obecny przynajmniej jeden przedstawiciel grupy i zaprezentować aktualną (działającą!) wersję programu.

### Punktacja

Za implementację całej funkcjonalności każdy student może otrzymać maksymalnie **100 punktów**. Poprawność kodu powinna być na bieżąco weryfikowana testami jednostkowymi napisanymi przy użyciu JUnit (**40 punktów**). Stopień pokrycia testami może zostać zweryfikowany przez prowadzącego np. za pomocą EclEmma. Na bieżąco wysyłaj sensownie opisane zmiany do repozytorium (**20 punktów**). Przy oddawaniu projektu powinieneś opisać jego strukturę i działanie istotnych elementów – stwórz w tym celu diagram klas i inne diagramy, które mogą być pomocne (**20 punktów**). Łącznie każdy student może otrzymać **180 punktów**.

Za każdy błąd w działaniu programu, który nie będzie wykryty w testach, a zostanie wykryty przez prowadzącego można stracić **10 punktów** (za listę można otrzymać ujemną liczbę punktów). Począwszy od 28 listopada nieobecność na zajęciach przedstawiciela grupy każdorazowo wiąże się z odjęciem **10 punktów** od wyniku końcowego. Oddanie projektu wymaga obecności obu członków zespołu.

Powodzenia, J.L.

---

<sup>2</sup>Zasada komi: rozpoczynające grę czarne mają większe szanse od białych. Aby gra była uczciwa można tę różnicę wyrównać doliczając białym dodatkowe punkty. Obecnie zwyczajowo dolicza się białym 6,5 punkta.