

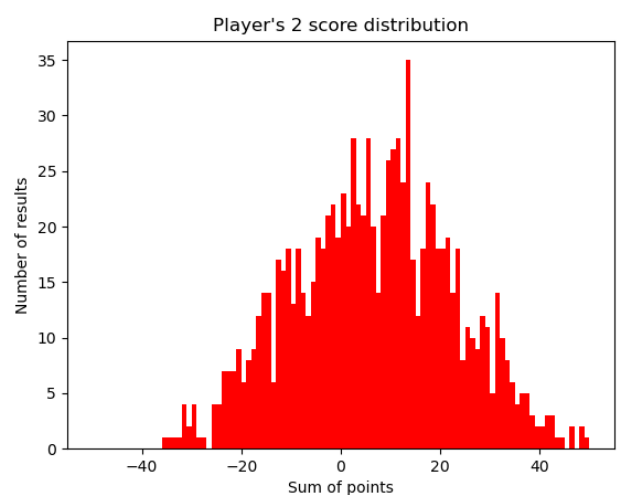
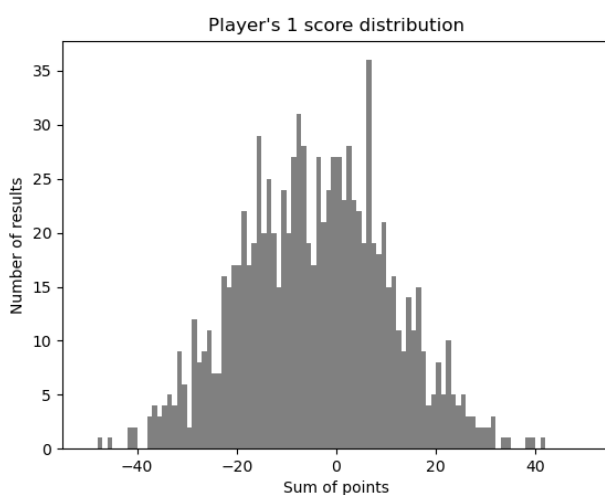
# WSI LAB. 3

Wyniki działania programu dla gry MinMax (gracz 1) vs Greedy Agent (gracz 2) w 1000 rozgrywkach dla  $n=15$

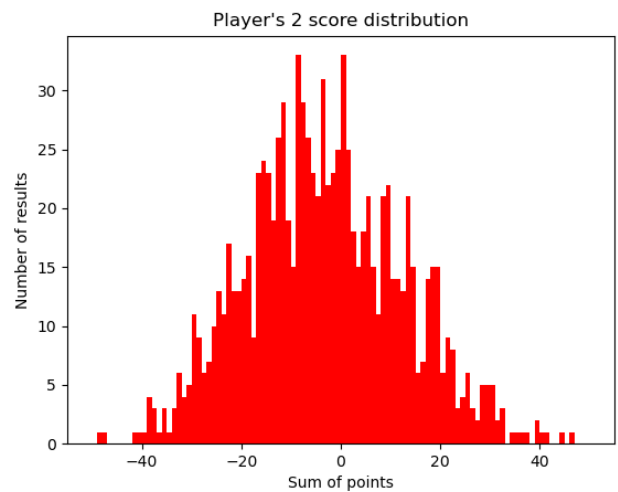
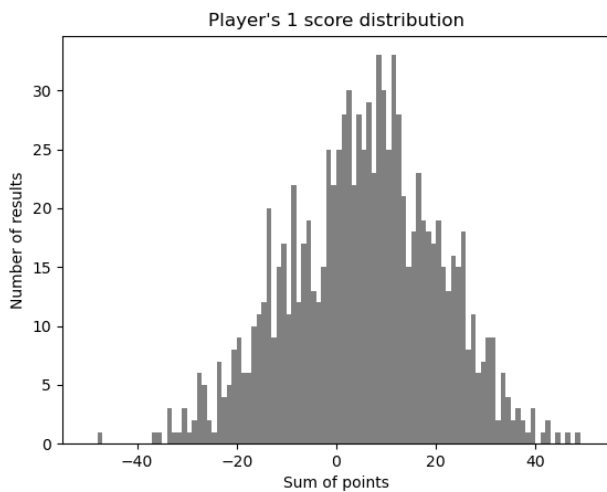
Wszystkie wartości w tabeli to średnie wyniki dla 1000 rozgrywek

Czas gry	Suma punktów gracz 1	Suma punktów gracz 2	Odchylenie standardowe sum punktów gracz 1	Odchylenie standardowe sum punktów gracz 2
Głębokość drzewa stanów = 1				
0,00002	0,43	0,29	15,52	15,67
Głębokość drzewa stanów = 5				
0,00024	5,00	-3,79	14,85	15,68
Głębokość drzewa stanów = 10				
0,004	5,07	-3,85	15,04	16,62
Głębokość drzewa stanów = 50				
0,019	5,01	-3,79	15,05	15,64

Histogram wygenerowane dla drzewa o głębokości = 1



Histogram wygenerowane dla drzewa o głębokości = 50

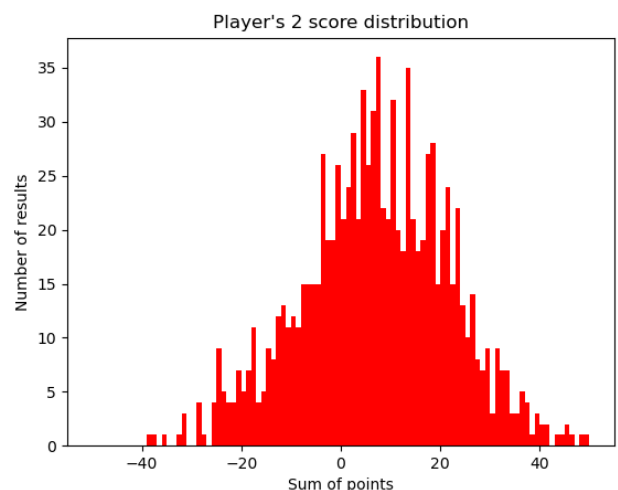
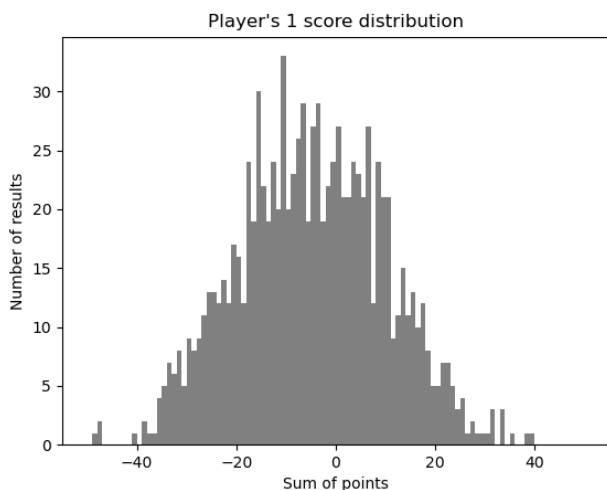


Wyniki działania programu dla gry MinMax (gracz 1) vs Ninja Agent (gracz 2) w 1000 rozgrywkach dla  $n=15$

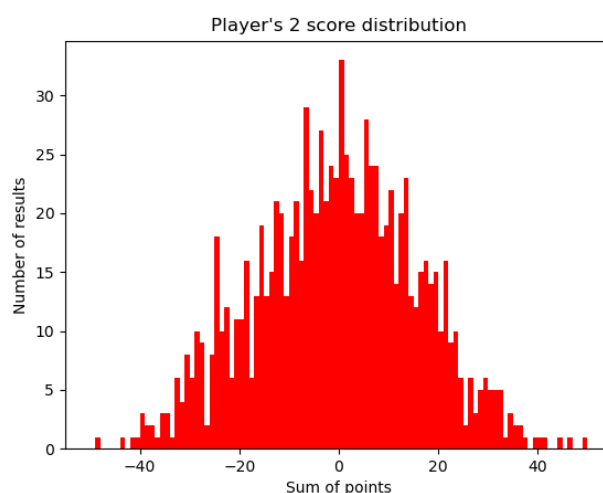
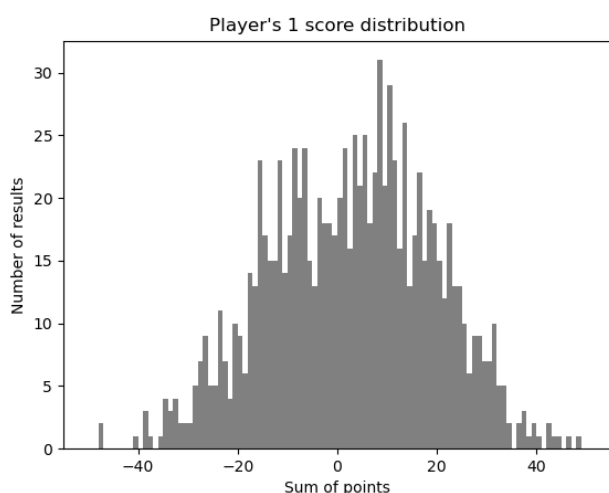
Wszystkie wartości w tabeli to średnie wyniki dla 1000 rozgrywek

Czas gry	Suma punktów gracz 1	Suma punktów gracz 2	Odchylenie standardowe sum punktów gracz 1	Odchylenie standardowe sum punktów gracz 2
Głębokość drzewa stanów = 1				
0,00003	-1,68	2,9	15,04	15,86
Głębokość drzewa stanów = 5				
0,0003	1,78	-0,56	14,42	16,24
Głębokość drzewa stanów = 10				
0,004	2,29	-1,07	15,54	15,2
Głębokość drzewa stanów = 50				
0,02	2,29	-1,07	14,56	15,18

Histogram wygenerowane dla drzewa o głębokości = 1



## Histogram wygenerowane dla drzewa o głębokości = 50

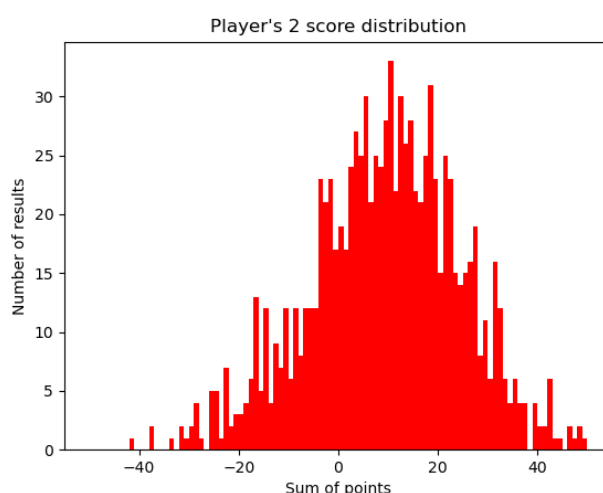
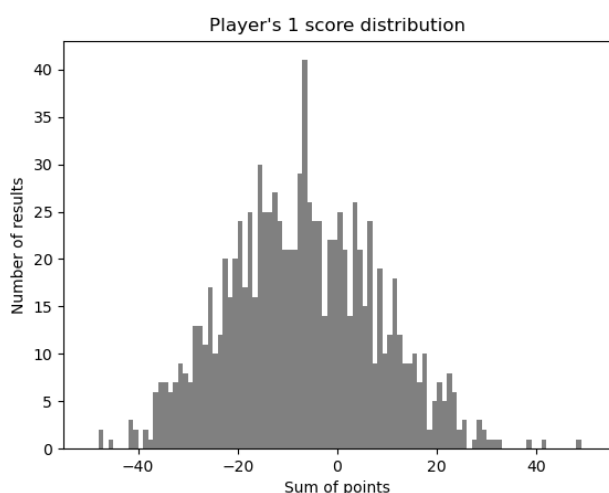


Wyniki działania programu dla gry MinMax (gracz 1) vs MinMax Agent o głębokości drzewa = 50 (gracz 2) w 1000 rozgrywkach dla  $n=15$

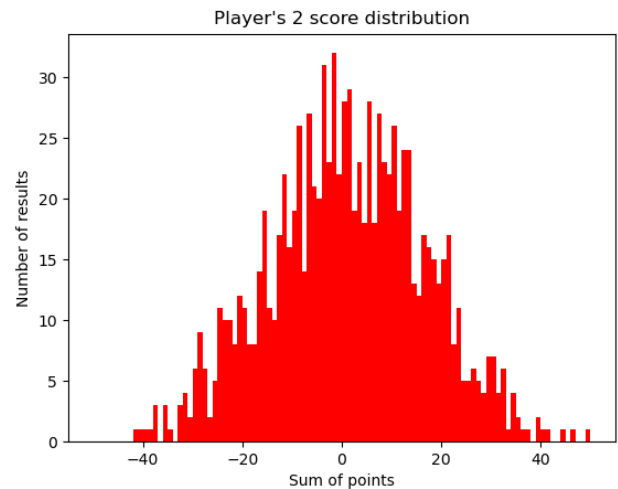
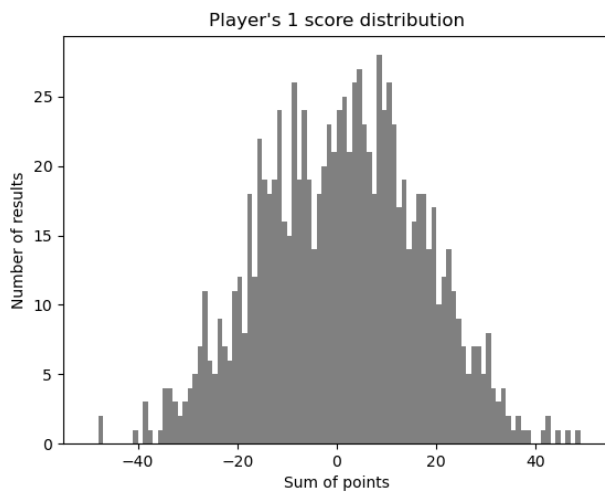
Wszystkie wartości w tabeli to średnie wyniki dla 1000 rozgrywek

Czas gry	Suma punktów gracz 1	Suma punktów gracz 2	Odchylenie standardowe sum punktów gracz 1	Odchylenie standardowe sum punktów gracz 2
Głębokość drzewa stanów = 1				
0,019	-4,1	5,32	15,11	14,06
Głębokość drzewa stanów = 5				
0,019	-0,21	1,42	12,91	12,54
Głębokość drzewa stanów = 10				
0,023	0,53	0,69	13,03	12,52
Głębokość drzewa stanów = 50				
0,039	0,55	0,66	13,03	12,52

## Histogramy wygenerowane dla drzewa o głębokości = 1



## Histogram wygenerowane dla drzewa o głębokości = 50



## Podsumowanie

### Jak głębokość drzewa wpływa na wyniki uzyskiwane przez MinMax Agent?

Głębokość drzewa usprawnia działanie MinMax Agent. Jeżeli głębokość drzewa jest równą 1 MinMax Agent zachowuje się dokładnie jak Greedy Agent, czyli wybiera większą z 2 możliwych opcji. Zwiększanie głębokości drzewa ulepsza działanie MinMax Agent do głębokości maksymalnej drzewa rozwiązań tzn., jeżeli zwiększymy głębokość drzewa na nieważne jak dużą wielkość większa niż wielkość pełnego drzewa rozwiązań nie wpłynie to na działanie MinMax Agent, ponieważ i tak rozwiąże on całe drzewo, na podstawie którego dokona wyboru.

### Jaki jest rozkład sumy punktów osiągnięty przez agenta MinMax?

Rozkład sumy punktów osiągniętych przez MinMax jest symetryczny względem średniej z uzyskanych przez niego punktów. Im większe odchylenie od średniej tym mniejsza ilość wyników.

### Jak na podstawie histogramu stwierdzić który agent jest lepszy?

Im bardziej wyniki na histogramie przesunięte są w lewą stronę tym gorzej poradził sobie dany agent. Jeżeli wyniki na histogramie plasują się po prawej stronie histogramu, oznacza to, że agent poradził sobie lepiej. W sytuacji MinMax vs MinMax o głębokości drzewa równej 50 histogram jest niemal symetryczny co pokazuje, że oba algorytmy radziły sobie podobnie.

### Jak przyspieszyć działanie MinMax agenta?

W celu przyspieszenia działania MinMax agenta można zastosować przycinanie alfa-beta. Algorytm ten przyspiesza wyszukiwanie MinMax dzięki eliminacji przypadków, które nie będą używane. Algorytm ten nie ocenia poddrzew których jakość jest niższa od już wcześniej ocenionych.

### Jak poprawić działanie MinMax agenta dla głębokości drzewa przeszukiwań równej 1?

Nie da się poprawić działania MinMax agenta dla głębokości równej 1 bez zwiększenia tej wartości. Agent ten będzie zachowywał się dokładnie tak jak Greedy Agent, który wybiera większą liczbę z 2 możliwych. W tym przypadku MinMax zachowuje się tak ponieważ nie widzi on żadnych innych rozwiązań poza 2 skrajnymi liczbami z lewej i z prawej strony, czyli nie może wziąć pod uwagę tego co znajduje się w środku, dlatego jedyną optymalną opcją jest wybranie większej z 2 możliwych do wybrania liczb.

