**Algorytmy związane z aktywnością bota**

Idea działania bota: Bot porusza się w sposób ciągły, nie zatrzymuje się kiedy nie jest to konieczne. Stanięcie w miejscu bota jest możliwe tylko w przypadku gdy będzie on otoczony ze wszystkich stron przez bloki zniszczalne/niezniszczalne. Ten przypadek jest bardzo mało prawdopodobny, choć możliwy (czerwone pole oznacza, że mapa w tym miejscu zmniejsza się):



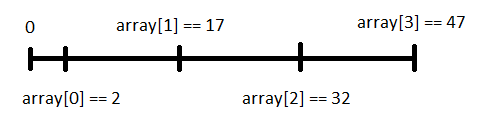
Bot w takim przypadku stanie w miejscu, z każdym wywołaniem funkcji \_physics\_process będzie miał 50% szans na podłożenie bomby (ponieważ znajduje się obok niego blok zniszczalny) i po zniszczeniu tego bloku (przez siebie lub innego gracza) kontynuuje poruszanie się (jeśli nie zginie od bomby).  
  
Stawianie bomby przez bota realizowane jest na 2 sposoby (oczywiście bot musi mieć dostępną jakąś bombę do podłożenia):

1. Bot po wykryciu, że podłożona przez niego bomba spowodowałaby zniszczenie przynajmniej 1 bloku ma 50% szans na postawienie bomby na aktualnej pozycji. Sprawdzane jest to przez pętlę wykonywaną 4 razy (w każdym z kierunków: prawo/lewo/góra/dół). Iteracyjnie sprawdzamy czy w odpowiednim kierunku znajduje się jakiś zniszczalny blok. Jeśli tak, zwracamy true (postawienie bomby ma sens). Jeśli nie, sprawdzamy w innym kierunku czy postawienie bomby ma sens. Jeśli w którymś z kierunków trafimy na blok niezniszczalny to zwracamy false (wybuch bomby zatrzymuje sie na blokach niezniszczalnych, nie ma sensu sprawdzać czy za nim jest jakiś blok zniszczalny). Bot musi stać na środku 64x64px pola planszy by postawić bombę (inaczej: może podłożyć bombę po wykonaniu pełnego ruchu w danym kierunku).  
   Za sprawdzenie, czy postawienie bomby ma sens odpowiada funkcja possible\_plant(position)
2. Bot ma 50% szans na podłożenie bomby w sposób losowy. Wprowadziliśmy to rozwiązanie, aby bot nie stawiał tylko bomb do zniszczania bloków zniszczalnych. W funkcji \_ready() wywoływana jest funkcja random\_planting() odpowiedzialna za losowe stawianie bomb przez bota. W tejże funkcji ustawiany jest licznik (Timer) na 1.5 sekundy. Po tym czasie bot ma 50% szans na postawienie bomby. Licznik tworzony jest na nowo, cały proces trwa aż do śmierci bota. Bot nie musi stać na środku 64x64px pola planszy by postawić bombę.

Poruszanie się bota realizowane jest w następujący sposób:

W momencie gdy bot dotarł do celu (ruch w prawo/lewo/górę/dół o 64px) wyznaczany jest kolejny kierunek poruszania się bota. Jeśli bot podłożył w tym momencie bombę (w sposób opisany w 1. powyżej) to zmienia kierunek ruchu na przeciwny (symuluje to uciekanie bota od miejsca podłożenia bomby). Jeśli bot nie postawił bomby (w sposób opisany w 1. powyżej) to ustalany jest nowy kierunek poruszania się bota według następującego algorytmu:

W funkcji new\_direction tworzona jest tablica z wartościami, które pomogą nam ustalić, w którym kierunku ma poruszyć się teraz bot.  
Tablica array symuluje dyskretną formę odcinka na osi x o początku w x=0 i końcu w x>=0 , x należy do liczb naturalnych. z liczbami typu int. Array[0] wskazuje na koniec 1 części odcinka od lewej strony. Array[0] jest więc liczbą naturalną >= 0. Analogicznie array[1] jest liczbą naturalną >= array[0].  
Idea przedstawiona na rysunku:



Ten przypadek oznacza, że bot stoi w miejscu z którego może iść w dowolym kierunku, a dotarł do tego miejsca poprzez ruch w lewo od poprzedniej pozycji (przed wykonaniem ruchu).

Idea: jeśli bot poruszył się w prawo, to cofnięcie się w lewo będzie następowało z mniejszym prawdopodobieństwem. Unikamy w ten sposób nienaturalnego sposobu poruszania się bota (np. częste ruchy prawo->lewo->prawo->lewo itp.).   
Po ustaleniu tablicy losujemy liczbę z zakresu [ 0, array[3] ) o typie integer. Wpadnięcie wylosowanej liczby do opowiedniej części odcinka powoduje określenie w którym kierunku ma poruszyć się teraz bot. Idąc od lewej strony odcinka mamy go podzielonego na części odpowiedzialne za ruch w prawo, lewo, górę oraz dół.

Jeśli array[3] == 0 tzn, że nie jest możliwy ruch w żadnym z kierunków. Wtedy zmienna stop = true. Dopóki bot nie wydostanie się z zablokowanego miejsca, nie może określić kolejnego kierunku poruszania się.

W momencie gdy bot wyliczył kierunek swojego następnego ruchu, aktualizowany jest wektor kierunku poruszania się bota i ustalana jest pozycja, którą musi osiągnąć (o 64 jednostki w prawo/lewo/górę/dół w zależności od wyznaczenia kierunku).

Dopóki bot nie osiągnie docelowego miejsca nie może postawić bomby w sposób opisany w punkcie 1.

Algorytm sprawdzania poprawności nicków

Algorytm działania bomby

Algorytm ustawiania klawiszy (sterowania)