



Witaj! W tej części dokumentacji znajdziesz informacje o wszystkich logikach, które występują w grze.











wersja: finalna, autor: Pejti

Podziękowania dla: Art00r, Zax (niestandardowe zastostawnie dla wybranych logik).

Spis treści

1. Logiki - krótka informacja.....	3
a) Logiki od A do E.....	4
b) Logiki od F do J	15
c) Logiki od K do O	21
d) Logiki od P do S.....	26
e) Logiki od T do Z	37
f) Niestandardowe zastosowanie logik	43

Logiki zostały oznaczone różnymi kolorami:

Kolor	Opis
	Podstawowe dekoracje
	Dodatkowe lub niestandardowe dekoracje
	Przeciwnicy
	Bossowie lub logiki z nimi powiązane
	Dźwięki
	Obiekty ruchome np. Elevatory/TogglePegi itp.
	Obiekty posiadające skarby/Powerupy np. skrzynia
	Skarby lub Powerupy
	Nie działająca logika lub powiązana z inną
	Obiekty raniące Clawa
	Pozostałe logiki

UWAGA!

W logice OneWayTriggerElevator, OneWayStartElevator oraz w logikach TriggerElevator, StartElevator i StopElevator minimalna prędkość z jaką może poruszać się platforma zależy od wartości FPS w grze. Np. jeżeli w grze masz 60FPS, to minimalna wartość prędkości dla platformy również wyniesie 60. Jeżeli wpisana wartość jest mniejsza od FPS to platforma będzie stać w miejscu.

1. Logiki - krótka informacja

Tworzenie obiektów i wpisywanie logik dla nich jest bardzo ważne. Zawsze gdy to robisz, rób to rozważnie i sprawdzaj więcej niż raz, czy aby na pewno niczego nie przeoczyłeś. Jeżeli po raz pierwszy masz styczność z logikami to staraj się je zapamiętać, aby w przyszłości dodawać obiekty szybciej i sprawniej.

Pamiętaj, że logiki możesz wykorzystać w każdym poziomie, ale nie w każdym będą działać prawidłowo. Ponadto nie każdy poziom ma przygotowaną grafikę dla wybranej logiki. Przed wstawieniem obiektu upewnij się, że jest przygotowana przez twórców grafika. Możesz to zrobić przeglądając już stworzony poziom (najlepiej Retail).

Attributes

Score: 0	Points: 0	Smarts: 0
Powerup: 0	Damage: 0	Health: 0
Speed X: 0	Speed Y: 0	Face Dir: 0
<input type="checkbox"/> X Min: 0	X Max: 0	Direction: 0
<input type="checkbox"/> Y Min: 0	Y Max: 0	Speed: 0

Other

Powyżej został pokazany obrazek, a na nim część okienka, które widzimy podczas dodawania/edytowania obiektu. Niektóre z obiektów wymagają uzupełnienia dodatkowych pól. Pod przyciskami **Rects...** i **Flags...** znajdziemy dodatkowe pola, które mogą się przydać.

Object Rectangles

☒ Move Rect
Left: 0 Top: 0 Right: 0 Bottom: 0

☒ Hit Rect
Left: 0 Top: 0 Right: 0 Bottom: 0

☒ Attack Rect
Left: 0 Top: 0 Right: 0 Bottom: 0

☒ Clip Rect
Left: 0 Top: 0 Right: 0 Bottom: 0

☒ User1 Rect
Left: 0 Top: 0 Right: 0 Bottom: 0

☒ User2 Rect
Left: 0 Top: 0 Right: 0 Bottom: 0

Object Flags

Dynamic Flags

☐ No Hit ☐ Always Active ☐ Safe
☐ Auto Hit Damage

Add Flags

☐ Difficult ☐ Eye Candy ☐ High Detail
☐ Multiplayer ☐ Extra Memory ☐ Fast CPU

Draw Flags

☐ No Draw ☐ Mirror ☐ Invert
☐ Flash

User Flags

☐ Flag 1 ☐ Flag 2 ☐ Flag 3
☐ Flag 4 ☐ Flag 5 ☐ Flag 6
☐ Flag 7 ☐ Flag 8 ☐ Flag 9
☐ Flag 10 ☐ Flag 11 ☐ Flag 12

U góry, po lewej stronie okienko Rects..., a po prawej stronie okienko Flags...

a) Logiki od A do E

Na następnych stronach znajduje się alfabetyczny spis logik wraz z opisem (pewne logiki zostały pominięte celowo):

2.1 **Logika AmbientPosSound** || NIE DZIAŁA lub jest częścią innej logiki


2.2 **Logika AmbientSound** || dźwięk np. łańcuchów/roślinności itp., który odtwarzany jest cały czas. Zasięg działania logiki można dowolnie ustalić.

Animation: wstawiamy nazwę dźwięku z tabeli dźwięków

XMin: ??, **XMax:** ?? - zasięg logiki w poziomie

YMin: ??, **YMax:** ?? - zasięg logiki w pionie

Damage: ?? - głośność dźwięku

Dodatkowo zaznaczamy *Always Active* i *No Draw* w .

Grafika



Nazwa grafiki

GAME_SOUNDICON

Domyślna wartość Z

9000

2.3 **Logika AmmoPowerup** || amunicja do pistoletu i dynamitu*

Smarts: ?? - możemy wpisać wartość np. 25 - tyle dostaniemy dynamitu*

Przykład grafiki



Nazwa grafiki

GAME_AMMO_SHOTBAG

Domyślna wartość Z

5000

2.4 Logika AmuletGemLogic || związana z amuletem/zdobywaniem kolejnych klejnotów*

2.5 **Logika AniCycle** || animowany obiekt

Animation: ?? - możemy wybrać prędkość animacji z folderu GAME/ANIS lub BASE poziomu np. GAME_CYCLE50

2.6 Logika **AniCycleNormal** || działa tak samo jak logika *AniCycle*, używana dla dużych grafik

Animation: ?? - możemy wybrać prędkość animacji z folderu GAME/ANIS lub BASE poziomu np. GAME_CYCLE50

2.7 **Logika AniRope** || liana/łańcuch

SpeedX: ?? - prędkość poruszania się liany/łańcucha, im wyższa wartość tym ruch jest wolniejszy. Jeżeli wpiszemy wartość 99 999, lina nie będzie się poruszać.

Przykład grafiki (Poziom3)



Nazwa grafiki


LEVEL_ROPE

Domyślna wartość Z


0

- 2.8 **Logika Aquatis** || szósty Boss, poziom XII. Posiada przy sobie zielony klejnot (domyślnie).

SpeedX: ??, SpeedY: ?? - współrzędne - w to miejsce poleci klejnot po pokonaniu Bossa. Jeżeli zostawimy puste pola, klejnot będzie traktowany jako dekoracja. Ten Boss stoi cały czas w miejscu.


Grafika	Nazwa grafiki	Domyślna wartość Z
	LEVEL_KINGAQUATIS	0

- 2.9 **Logika AquatisCrack** || dziura do której należy wrzucić dynamit, aby aktywować spadające stalaktyty

Grafika	Nazwa grafiki	Domyślna wartość Z
	LEVEL_AQUATISWALLCRACK	-10

- 2.10 **Logika AquatisDynamite** || dynamit, który odnawia się po jego wykorzystaniu podczas walki z szóstym Bossem

XMin: -500, XMax: 1250, YMin: -300, YMax: 500 - domyślne wartości

Grafika	Nazwa grafiki	Domyślna wartość Z
	GAME_DYNAMITE	0

- 2.11 **Logika AquatisStalactite** || stalaktyt, który spada i rani szóstego Bossa. Wymaga aktywacji logiki *AquatisCrack*.

Smarts: ?? - wartość określa kolejność w spadaniu np. wpisanie 1 oznacza, że właśnie ten stalaktyt spadnie na Bossa jako pierwszy

Dodatkowo musimy wypełnić pola **Move Rect: ??, ??, ??, ??, Hit Rect: ??, ??, ??, ??**, oraz **Attack Rect: ??, ??, ??, ??** aby stalaktyt działał prawidłowo (ustalenie obszaru na którym znajduje się Boss). Każde pole *Left: ??, Top: ??, Right: ??, Bottom: ??*, musi posiadać tę samą wartość.

Move Rect			
Left: 37791	Top: 3519	Right: 37825	Bottom: 3620
Hit Rect			
Left: 37791	Top: 3519	Right: 37825	Bottom: 3620
Attack Rect			
Left: 37791	Top: 3519	Right: 37825	Bottom: 3620

Przykład (Retail12):

Grafika

Nazwa grafiki

Domyślna wartość Z



LEVEL_PROJECTILES_STALACTITE

1000

- 2.12 **Logika BackgroundLogic** || obiekt z tą logiką będzie podążać za Clawem
SpeedX: ??, **SpeedY**: ?? - wartość prędkości w poziomie i w pionie, wartość 100 odpowiada prędkości poruszania się Clawa
- 2.13 **Logika BackStackedCrates** || skrzynie poustawiane jedna na drugiej stojące z tyłu.
Wystarczy wstawić jedną, a później wypełnić pola:
Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny
User1 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów
User2 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów
Im więcej wypełnimy pól tym więcej skrzyń zostanie wstawionych (maks=9).

Przykład grafiki (Poziom1)

Nazwa grafiki

Domyślna wartość Z



LEVEL_CRATES

1000

- 2.14 **Logika BearSailor** || niedźwiedź-marynarz. Występuje w kilku poziomach.
Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny
User1 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów
User2 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów
Maksymalna ilość skarbów to 9.
XMin: ??, **XMax**: ?? - jest to obszar w poziomie dla patrolującego przeciwnika
Zawsze XMin < XMax np. XMin: 1000, XMax: 1500.

Przykład grafiki (Poziom7)

Nazwa grafiki

Domyślna wartość Z



LEVEL_BEARSAILOR

0

- 2.15 **Logika BehindAniCandy** || dekoracja, która jest animowana - włączenie niskich detali w opcjach gry spowoduje, że obiekt z tą logiką nie będzie wyświetlany

Przykład grafiki (Poziom1)



Nazwa grafiki
LEVEL_TORCH

Domyślna wartość Z
1000

- 2.16 **Logika BehindCandy** || dekoracja - włączenie niskich detali w opcjach gry spowoduje, że obiekt z tą logiką nie będzie wyświetlany

Przykład grafiki (Poziom1)



Nazwa grafiki
LEVEL_SKULL

Domyślna wartość Z
1000

- 2.17 **Logika BehindCrate** || skrzynia stojąca z tyłu

Health: 2 - trzeba uderzyć skrzynię 2 razy, aby ją zniszczyć

Health: 999 - nie będzie możliwe zniszczenie skrzyni - lepiej nie używać

Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny

Przykład grafiki (Poziom1)



Nazwa grafiki
LEVEL_CRATES

Domyślna wartość Z
1000

- 2.18 **Logika BehindStatue** || statua zawierająca skarby (poziom5)

Health: 2 - trzeba uderzyć statuetkę 2 razy, aby ją zniszczyć

Health: 999 - nie będzie możliwe zniszczenie statui - lepiej nie używać

Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny

User1 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

User2 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

Grafika



Nazwa grafiki

LEVEL_STATUE

Domyślna wartość Z

1000/1010

- 2.19 **Logika BigSoundTrigger** || odtwarzanie dźwięku np. spadających kropeł lub jednorazowe dźwięki, które są odtwarzane kiedy wejdziemy w obszar logiki (zasięg 125 pikseli)

Animation: wstawiamy nazwę dźwięku z tabeli dźwięków

Smarts: -1 - dźwięk będzie odtwarzany za każdym razem, gdy wejdziemy w jego obszar działania lub zostawiamy 0 - dźwięk odtworzy się tylko raz, wpisanie wartości większej od zera oznacza ile razy możemy "uruchomić" dźwięk, logika ta resetowana jest przy każdej śmierci Clawa w zależności od wartości *Smarts*

Dodatkowo zaznaczamy *Always Active* i *No Draw* w **Flags...**

Grafika



Nazwa grafiki

GAME_SOUNDICON

Domyślna wartość Z

9000

2.20 **Logika BlinkingEyes** || obiekt z tą logiką znika po 1,5s

2.21 Logika BossHealthMeter || pasek zdrowia Bossa

2.22 **Logika BossStager** || po wejściu w obszar działania tej logiki, Claw przeprowadza dialog z Bossem. Pojawia się, a później znika duża grafika ukazująca napis "Defeat 'nazwa bossa'".

Dodatkowo zaznaczamy *No Draw* w .

Grafika



Nazwa grafiki

GAME_SOUNDICON

Domyślna wartość Z

8000

2.23 Logika BossStagerAttack || Rect dla logiki BossStager

2.24 **Logika BossWarp** || teleport, przenosi nas do Bossa w danym poziomie
SpeedX: ??, **SpeedY**: ?? - wpisujemy współrzędne miejsca, w które zostaniemy przeniesieni

Grafika



Nazwa grafiki

GAME_BOSSWARP

Domyślna wartość Z

1000

2.25 **Logika BouncingGoodie** || obiekt z tą logiką może poruszać się i jednocześnie spełniać swoją własną logikę np. poruszający się skarb czy Powerup, przy kontakcie ze ścianą/podłożem zmniejsza się prędkość obiektu

Smarts: ?? - czas (wartość) dla powerupów w ms (milisekundy), 1000ms = 1s

SpeedX: ??, **SpeedY**: ?? - prędkość w poziomie i w pionie oraz kąt pod jakim obiekt zacznie się poruszać względem podłoża. Kąt można odpowiednio dobrać posługując się tabelami dla logiki *SkullCannon*.

2.26 **Logika BreakPlank** || łamiące się pod Clawem deski (poziom5), a także pękające kawałki skały (poziom11)

Counter: ?? - czas (wartość) po jakim deski/skały zaczną się kruszyć, podajemy w ms (milisekundy), 1000ms = 1s

Width: ?? - szerokość obiektu, wpisanie np. liczby 2 będzie oznaczać, że pojawią się dodatkowe 2 obiekty tj. ostatecznie będziemy mieć 3 deski/skały obok siebie

UWAGA - w miejscu gdzie wstawiamy obiekt z tą logiką musimy dodatkowo wstawić kafelki o nr 509 (poziom5) lub kafelki o nr 39 (poziom11) tak, aby pokrywał się z obiektem. Jeżeli chcemy wstawić 5 kruszących się desek/skał to musimy również wstawić 5 kafelków.

Atrybuty *Counter* i *Width* wypełniamy wartością w w ustawieniach obiektu.

Przykład grafiki (Poziom5)



Nazwa grafiki

LEVEL_BREAKPLANK


Domyślna wartość Z

1010


2.27 **Logika Bullet** || obiekt, który porusza się i jeżeli wpadnie na kafelek o atrybucie: Ground/Solid/Climb to znika
SpeedX: ??, **SpeedY**: ?? - prędkość w poziomie i w pionie oraz kąt pod jakim obiekt zacznie się poruszać względem podłoża. Kąt można odpowiednio dobrać posługując się tabelami dla logiki *SkullCannon*.

2.28 Logika CannonBall || pocisk, używany przez inną logikę

2.29 **Logika CannonButton** || przycisk, który musimy uderzyć (przełączyć) podczas walki z Gabrielem

Grafika	Nazwa grafiki	Domyślna wartość Z
	LEVEL_CANNONBUTTON	1000

2.30 **Logika CannonSwitch** || dźwignia, którą przełącza Gabriel podczas walki

Grafika	Nazwa grafiki	Domyślna wartość Z
	LEVEL_CANNONSWITCH	1000


2.31 Logika CaptainClaw || jest to Kapitan Pazur czyli postać, którą gramy. Lepiej nie wstawiać do poziomu.

2.32 Logika CaptainClawHit ||

2.33 Logika CaptainClawRemoteRacer ||

2.34 Logika CaptainClawScreenPosition ||

2.35 **Logika Chameleon** || kameleon. Występuje w poziomie XIII i XIV.
Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny
User1 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów
User2 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów
Maksymalna ilość skarbów to 9.
XMin: ??, **XMax**: ?? - jest to obszar w poziomie dla patrolującego przeciwnika
Zawsze XMin < XMax np. XMin: 1000, XMax: 1500.

Grafika	Nazwa grafiki	Domyślna wartość Z
	LEVEL_CHAMELEON	0

2.36 Logika Checkpoint || zwykły punkt zapisu, często stawiany w poziomie

Grafika



Nazwa grafiki

GAME_CHECKPOINTFLAG

Domyślna wartość Z

1000

2.37 Logika CheckpointAttack || Rect dla logiki Checkpoint

2.38 Logika CheckpointTimer ||

2.39 Logika ChildChar ||

2.40 Logika ChildLight ||

2.41 Logika ChildRibbon ||

2.42 Logika ChildScoreNum ||

2.43 Logika ClawDialogBigSoundTrigger || dialog Clawa. Zasięg działania logiki wynosi 125 pikseli.

Animation: wstawiamy nazwę dialogu z tabeli dialogów

Smarts: wpisanie wartości większej od zera oznacza ile razy możemy "uruchomić" dialog. Logika ta resetowana jest przy każdej śmierci Clawa w zależności od wartości *Smarts* (0 - zawsze się zresetuje).

Dodatkowo zaznaczamy *Always Active* i *No Draw* w .

Grafika



Nazwa grafiki

GAME_SOUNDICON

Domyślna wartość Z

9000

2.44 Logika ClawDialogHugeSoundTrigger || dialog Clawa. Zasięg działania logiki wynosi 255 pikseli.

Animation: wstawiamy nazwę dialogu z tabeli dialogów

Smarts: wpisanie wartości większej od zera oznacza ile razy możemy "uruchomić" dialog. Logika ta resetowana jest przy każdej śmierci Clawa w zależności od wartości *Smarts* (0 - zawsze się zresetuje).

Dodatkowo zaznaczamy *Always Active* i *No Draw* w .

Grafika



Nazwa grafiki

GAME_SOUNDICON

Domyślna wartość Z

9000

2.45 Logika ClawDialogSmallSoundTrigger || dialog Clawa. Zasięg działania logiki wynosi 30 pikseli.

Animation: wstawiamy nazwę dialogu z tabeli dialogów

Smarts: wpisanie wartości większej od zera oznacza ile razy możemy "uruchomić" dialog. Logika ta resetowana jest przy każdej śmierci Clawa w zależności od wartości *Smarts* (0 - zawsze się zresetuje).

Dodatkowo zaznaczamy *Always Active* i *No Draw* w .



Nazwa grafiki
GAME_SOUNDICON

Domyślna wartość Z
9000

2.46 **Logika ClawDialogSound** || NIE DZIAŁA lub jest częścią innej logiki

2.47 **Logika ClawDialogSoundTrigger** || dialog Clawa. Zasięg działania logiki wynosi 60 pikseli.

Animation: wstawiamy nazwę dialogu z tabeli dialogów

Smarts: wpisanie wartości większej od zera oznacza ile razy możemy "uruchomić" dialog. Logika ta resetowana jest przy każdej śmierci Clawa w zależności od wartości *Smarts* (0 - zawsze się zresetuje).

Dodatkowo zaznaczamy *Always Active* i *No Draw* w .



Nazwa grafiki
GAME_SOUNDICON

Domyślna wartość Z
9000

2.48 **Logika ClawDialogTallSoundTrigger** || dialog Clawa. Zasięg działania logiki w poziomie wynosi 25 pikseli, a w pionie 95 pikseli.

Animation: wstawiamy nazwę dialogu z tabeli dialogów

Smarts: wpisanie wartości większej od zera oznacza ile razy możemy "uruchomić" dialog. Logika ta resetowana jest przy każdej śmierci Clawa w zależności od wartości *Smarts* (0 - zawsze się zresetuje).

Dodatkowo zaznaczamy *Always Active* i *No Draw* w .



Nazwa grafiki
GAME_SOUNDICON

Domyślna wartość Z
9000

2.49 **Logika ClawDialogTinySoundTrigger** || dialog Clawa. Zasięg działania logiki wynosi zaledwie 10 pikseli.

Animation: wstawiamy nazwę dialogu z tabeli dialogów

Smarts: wpisanie wartości większej od zera oznacza ile razy możemy "uruchomić" dialog. Logika ta resetowana jest przy każdej śmierci Clawa w zależności od wartości *Smarts* (0 - zawsze się zresetuje).

Dodatkowo zaznaczamy *Always Active* i *No Draw* w .



Nazwa grafiki
GAME_SOUNDICON

Domyślna wartość Z
9000

2.50 **Logika ClawDialogWideSoundTrigger** || dialog Clawa. Zasięg działania logiki w poziomie wynosi 95 pikseli, a w pionie 25 pikseli.

Animation: wstawiamy nazwę dialogu z tabeli dialogów

Smarts: 0 - wpisanie wartości większej od zera oznacza ile razy możemy "uruchomić" dialog. Logika ta resetowana jest przy każdej śmierci Clawa w zależności od wartości *Smarts* (0 - zawsze się zresetuje).

Dodatkowo zaznaczamy *Always Active* i *No Draw* w .




Nazwa grafiki
GAME_SOUNDICON

Domyślna wartość Z
9000

2.51 Logika ClawMultiRacer ||

2.52 Logika CoinPowerup || pieniędzy. Nie ma animacji *GLITTER* (połysk).

Grafika	Nazwa grafiki	Domyślna wartość Z
	GAME_TREASURE_COINS	1000

2.53 Logika ConveyorBelt || taśma z poziomu IX. Działa w obie strony.

Speed: ?? - wpisujemy wartość dodatnią (przesuwa Clawa w prawo) lub ujemną (przesuwa Clawa w lewo)

Grafika	Nazwa grafiki	Domyślna wartość Z
	LEVEL_CONVEYMIDDLE	1000

2.54 Logika CrabBomb || używana przez kraby rzucające bombami w poziomie VII

2.55 Logika CrabNest || gniazdo z którego wylatują kraby

W w **User 1:** ?? - wpisujemy liczbę krabów, od tego zależy ile ich wyleci kiedy zbliżymy się do gniazda (maks=9, min=1, domyślna wartość 0=3 kraby). Można też ustawić 0 krabów - należy wpisać wartość -1.

Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny

User1 Rect: ??, ??, ??, ?? - wartości dla skarbów (tabela)

User2 Rect: ??, ??, ??, ?? - wartości dla skarbów (tabela)

UWAGA - jeżeli w gnieździe jest np. 5 krabów to mamy cztery możliwości do wyboru odnośnie skarbów:

I - nie wypełniamy pól - żaden z krabów nie będzie mieć skarbu

II - wypełniamy tyle pól ile jest krabów - każdy dostanie po jednym

III - wypełniamy mniej pól niż mamy krabów np. 2 - dwa kraby dostaną po jednym skarbie, reszta krabów nie będzie posiadać skarbu

IV - wypełniamy więcej pól niż mamy krabów - każdy krab będzie mieć skarb + dodatkowe skarby wylecą z gniazda

XMin: ??, **XMax:** ??, **YMin:** ??, **YMax:** ?? - współrzędne obszaru, w który należy wejść, aby aktywować logikę

Grafika	Nazwa grafiki	Domyślna wartość Z
	LEVEL_SANDHOLE	0

2.56 Logika Crate || NIE DZIAŁA lub jest częścią innej logiki

2.57 Logika CrateHit || Rect dla skrzyni

2.58 Logika CratePowerup || odpowiedzialna za skarb w skrzyni

2.59 Logika CrazyHook || przeciwnik nr 2 w poziomie IX i X (inny kolor)

Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny

User1 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

User2 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

Maksymalna ilość skarbów to 9.

XMin: ??, **XMax:** ?? - jest to obszar w poziomie dla patrolującego przeciwnika

Zawsze XMin < XMax np. XMin: 1000, XMax: 1500.

Przykład grafiki (Poziom9)



Nazwa grafiki

LEVEL_CRAZYHOOK

Domyślna wartość Z

5000

- 2.60 **Logika CrumblingPeg** || klocek, który się rozsypuje kiedy na niego wskoczymy, odnawia się po śmierci Clawa

SpeedX: ?? - wartość w ms (milisekundy), możemy opóźnić rozsypanie się klocka wpisując np. 3000 - klocek rozsypie się dopiero po 3 sekundach

Przykład grafiki (Poziom1)



Nazwa grafiki

LEVEL_CRUMBLINGPEG

Domyślna wartość Z

0

- 2.61 **Logika CrumblingPegNoRespawn** || klocek, który się rozsypuje kiedy na niego wskoczymy, nie odnawia się po śmierci Clawa

SpeedX: ?? - wartość w ms (milisekundy), możemy opóźnić rozsypanie się klocka wpisując np. 3000 - klocek rozsypie się dopiero po 3 sekundach

Przykład grafiki (Poziom1)



Nazwa grafiki

LEVEL_CRUMBLINGPEG

Domyślna wartość Z

0

- 2.62 **Logika CursePowerup** || klątwa. Istnieje kilka rodzajów. W zależności od wybranego obrazka zadziała inaczej na pozostałych graczy.

Przykład grafiki



Nazwa grafiki

GAME_CURSES_TREASURE

Domyślna wartość Z

0

- 2.63 **Logika CutThroat** || przeciwnik nr 2 w poziomie III i IV (inny kolor)

Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny

User1 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

User2 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

Maksymalna ilość skarbów to 9.

XMin: ??, **XMax:** ?? - jest to obszar w poziomie dla patrolującego przeciwnika

Zawsze XMin < XMax np. XMin: 1000, XMax: 1500.

Przykład grafiki (Poziom3)



Nazwa grafiki

LEVEL_CUTTHROAT

Domyślna wartość Z

0

2.64 **Logika DoNothing** || pozostałe dekoracje (innych niż ...Candy) - nie mają animacji

Przykład grafiki (Poziom1)



Nazwa grafiki

LEVEL_ARROWSIGN_RIGHT

Domyślna wartość Z

1000

2.65 **Logika DoNothingNormal** || działa jak logika DoNothing - ma większą wartość Z (nie zawsze). Używana dla dużych grafik.

Przykład grafiki (Poziom4)



Nazwa grafiki

LEVEL_BUSH

Domyślna wartość Z

1000/5000/8000/8500/9000

2.66 **Logika Dynamite** || spadający dynamit. Wybucho po takim samym czasie jak zwykły.

Grafika



Nazwa grafiki

GAME_DYNAMITELIT

Domyślna wartość Z

0

2.67 **Logika Elevator** || poruszająca się platforma (można też ustawić zerową prędkość), można ustawić zasięg poruszania się w poziomie, w pionie lub połączyć pierwszą i drugą możliwość

XMin: ??, **XMax:** ?? - zasięg w poziomie ($XMin < XMax$)

YMin: ??, **YMax:** ?? - zasięg w pionie ($YMin < YMax$)

SpeedX: ?? - wartość dla prędkości w poziomie

SpeedY: ?? - wartość dla prędkości w pionie

Przykład grafiki (Poziom1)



Nazwa grafiki

LEVEL_ELEVATORS

Domyślna wartość Z

2000

2.68 **Logika EndLevelGem** || klejnot (koniec poziomu) - grafika zmieni się jeżeli zagramy w Custom Level lub w Multiplayerze

Grafika (taka się pojawi)




Nazwa grafiki

GAME_MAPPIECE
LEVEL_GEM

Domyślna wartość Z

0

- 2.69 **Logika EndOfLevelPowerup** || mapa/klejnot (koniec poziomu) - grafika zmienia się jeżeli zagramy w Custom Level lub w Multiplayerze

Grafika (taka się pojawi)	Nazwa grafiki	Domyślna wartość Z
	GAME_MAPPIECE LEVEL_GEM	0


- 2.70 Logika EnemyDialogSound || odpowiedzialna za odgłosy/dialogi przeciwników

b) Logiki od F do J


- 2.71 Logika FallingDebris ||

- 2.72 **Logika FireBullet** || pocisk ognisty, który może poruszać się, ale nie musi (np. animowana dekoracja) i jeżeli wpadnie na kafelek o atrybucie: Ground/Solid/Climb to znika, dodatkowy efekt - poświata

SpeedX: ??, **SpeedY**: ?? - prędkość w poziomie/w pionie oraz kąt pod jakim obiekt zacznie się poruszać względem podłoża. Kąt można odpowiednio dobrać posługując się tabelami dla logiki *SkullCannon*. Działa podobnie jak logika Bullet.


Grafika	Nazwa grafiki	Domyślna wartość Z
	GAME_PROJECTILES_FIRESWORD	0

- 2.73 Logika FirstSuperCheckpoint || pierwszy punkt zapisu

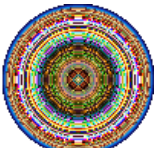
Grafika	Nazwa grafiki	Domyślna wartość Z
	GAME_SUPERCHECKPOINT	1000

- 2.74 **Logika Fish** || ryba, która występuje w poziomie XI i XII


XMin: ??, **XMax**: ?? - jest to obszar w poziomie dla pływającego przeciwnika
Zawsze XMin < XMax np. XMin: 1000, XMax: 1500.

Grafika	Nazwa grafiki	Domyślna wartość Z
	LEVEL_FISH	5000


- 2.75 **Logika Flare** || może służyć jako dodatkowe oświetlenie (dekoracja). Działa jak lampa. Zbyt duża ilość w poziomie może spowodować obniżenie FPSów.

Grafika	Nazwa grafiki	Domyślna wartość Z
	GAME_LIGHTFX_FLARE	0


- 2.76 **Logika FloorSpike** || wychodzący z ziemi kolec
SpeedX: ??, **SpeedY**: ?? - czas na jaki się wysunie (X) oraz czas na jaki się wsunie (Y). Można też nic nie wpisywać, zadziała wtedy z domyślnym czasem.

Przykład grafiki (Poziom3)	Nazwa grafiki	Domyślna wartość Z
	LEVEL_FLOORSPIKES1	0


- 2.77 **Logika FloorSpike2** || wychodzący z ziemi kolec
SpeedX: ??, **SpeedY**: ?? - czas na jaki się wysunie (X) oraz czas na jaki się wsunie (Y). Można też nic nie wpisywać, zadziała wtedy z domyślnym czasem.

Przykład grafiki (Poziom3)	Nazwa grafiki	Domyślna wartość Z
	LEVEL_FLOORSPIKES1	0

- 2.78 **Logika FloorSpike3** || wychodzący z ziemi kolec
SpeedX: ??, **SpeedY**: ?? - czas na jaki się wysunie (X) oraz czas na jaki się wsunie (Y). Można też nic nie wpisywać, zadziała wtedy z domyślnym czasem.


Przykład grafiki (Poziom3)	Nazwa grafiki	Domyślna wartość Z
	LEVEL_FLOORSPIKES1	0

- 2.79 **Logika FloorSpike4** || wychodzący z ziemi kolec
SpeedX: ??, **SpeedY**: ?? - czas na jaki się wysunie (X) oraz czas na jaki się wsunie (Y). Można też nic nie wpisywać, zadziała wtedy z domyślnym czasem.


Przykład grafiki (Poziom3)	Nazwa grafiki	Domyślna wartość Z
	LEVEL_FLOORSPIKES1	0

Twórcy gry stworzyli podobnie jak w przypadku TogglePegów cztery logiki dla koleców, które wysuwają się po kolei. Warto mieć to na uwadze przy tworzeniu poziomów z wykorzystaniem tych logik.

- 2.80 **Logika FrontAniCandy** || dekoracja, która jest animowana - włączenie niskich detali w opcjach gry spowoduje, że obiekt z tą logiką nie będzie wyświetlany

Przykład grafiki (Poziom1)	Nazwa grafiki	Domyślna wartość Z
	LEVEL_TORCHSTAND	5000

- 2.81 **Logika FrontCandy** || dekoracja - włączenie niskich detali w opcjach gry spowoduje, że obiekt z tą logiką nie będzie wyświetlany

Przykład grafiki (Poziom1)	Nazwa grafiki	Domyślna wartość Z
	LEVEL_WEB1	5000

2.82 **Logika FrontCrate** || skrzynia stojąca z przodu

Health: 2 - trzeba uderzyć skrzynię 2 razy, aby ją zniszczyć

Health: 999 - nie będzie możliwe zniszczenie skrzyni - lepiej nie używać

Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny

Przykład grafiki (Poziom1)



Nazwa grafiki

LEVEL_CRATES

Domyślna wartość Z

5000

2.83 **Logika FrontStackedCrates** || skrzynie poustawiane jedna na drugiej stojące z przodu, wystarczy wstawić jedną, a później wypełnić pola:

Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny

User1 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

User2 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

Im więcej wypełnimy pól tym więcej skrzyń zostanie wstawionych (maks=9).

Przykład grafiki (Poziom1)



Nazwa grafiki

LEVEL_CRATES

Domyślna wartość Z

5000

2.84 **Logika FrontStatue** || statua zawierająca skarby (poziom5)

Health: 2 - trzeba uderzyć statuę 2 razy, aby ją zniszczyć

Health: 999 - nie będzie możliwe zniszczenie statuy - lepiej nie używać

Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny

User1 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

User2 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

Grafika



Nazwa grafiki

LEVEL_STATUE

Domyślna wartość Z

5000


2.85 **Logika Gabriel** || czwarty Boss, poziom VIII. Posiada przy sobie czerwony klejnot (domyślnie).

Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny

User1 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów


User2 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

SpeedX: ??, SpeedY: ?? - współrzędne - w to miejsce poleci klejnot po pokonaniu Bossa. Jeżeli zostawimy puste pola, klejnot będzie traktowany jako dekoracja. Ten Boss stoi cały czas w miejscu.

Grafika	Nazwa grafiki	Domyślna wartość Z
	LEVEL_GABRIEL	2000

- 2.86 **Logika GabrielBomb** || używana przez Gabriela, kiedy z nim walczymy. Może być jednorazową przeszkodą w innym poziomie. Odbiera 5HP, zasięg ~50px.
SpeedX: ??, SpeedY: ?? - ustalamy prędkość oraz kąt pod jakim pocisk zacznie się poruszać. Początkowy kąt możemy ustalić jak w przypadku logiki *SkullCannon*. Pocisk odbija się od sufitu z tą samą prędkością. Odbija się również od ścian, ale wtedy zmniejsza mu się prędkość. Nie odbija się od podłoża do góry.

- 2.87 **Logika GabrielCannon** || działo Gabriela. Strzela do nas podczas walki.

Grafika	Nazwa grafiki	Domyślna wartość Z
	LEVEL_GABRIELCANNON	1000

- 2.88 **Logika GlitterBaby** || jednorazowa animacja *SPARKLE* (blask)

- 2.89 **Logika GlitterlessPowerup** || logika dla skarbu bez animacji *GLITTER* (połysk). Zawsze wykorzystywana dla monet.

Grafika	Nazwa grafiki	Domyślna wartość Z
	GAME_TREASURE_COINS	1000

- 2.90 Logika GlitterMother ||

- 2.91 **Logika GlobalAmbientSound** || dźwięk, który odtwarzany jest przez cały czas w każdym miejscu w poziomie. Dodatkowo można określić jego głośność, a nawet ustawić losowe odtwarzanie.

Damage: ?? - głośność dźwięku

W **Rects...** możemy ustawić losowość dla dźwięku:

Move Rect Left: ??, Top: ?? - ustawiamy na ile sekund ma się włączyć dźwięk

Move Rect Right: ??, Bottom: ?? - ustawiamy na ile sekund ma się wyłączyć dźwięk

Wartość podajemy w tysiącach: 1000 = 1s.

Smarts: ??

Przykład:

Move Rect Left: 3000, Top: 5000, Right: 10000, Bottom: 25000

Oznacza to, że dźwięk włączy się na 3 do 5 sekund, a wyłączy na 10 do 25 sekund.

Dodatkowo zaznaczamy *Always Active* i *No Draw* w **Flags...**.

Grafika

Nazwa grafiki
GAME_SOUNDICONDomyślna wartość Z
90002.92 **Logika GoldPowerup** || sztabka złota. Bez animacji *GLITTER* (połysk).

Grafika

Nazwa grafiki
GAME_TREASURE_GOLDBARSDomyślna wartość Z
10002.93 **Logika GooBubble** || animowany obiekt, który zasłania wodę/lawę etc.2.94 **Logika GooCoverup** || dla obiektu (grafiki), który ma za zadanie zasłaniać obszar poniżej wody/bagna/lawy itp., tzn. kafelki od których Claw ginie

Przykład grafiki



Nazwa grafiki

LEVEL_GOOCOVERUP

Domyślna wartość Z

5000/9000

2.95 **Logika GooVent** || obiekt z tą logiką pojawia się w poziomie VI i rani gracza - z kanałów wypływają ścieki na kilka sekund po czym z powrotem do niego wpływają
Damage: ?? - wpisujemy wartość dla zadawanych obrażeń (domyślnie ścieki ranią gracza i odbierają mu 1 punkt zdrowia)
Counter: ?? - czas (wartość) między kolejnym działaniem obiektu, w ms (milisekundy), 1000ms = 1s

Grafika

Nazwa grafiki
LEVEL_GOOVENTDomyślna wartość Z
0

2.96 **Logika GroundBlower** || obiekt, który wystrzeliwuje Clawa do góry
YMax: ?? - 0 wartość domyślna = 450, można również wpisać własną wartość np. 300 tzn. Claw wyleci 300 pikseli do góry.

Przykład grafiki (Poziom6)



Nazwa grafiki

LEVEL_BLOWING1

Domyślna wartość Z

5000




2.97 Logika GroundCannon ||

2.98 Logika HealthFrame || animacja zdrowia

2.99 **Logika HealthPowerup** || jedzenie (leczy 5 HP)/fiolka medyczna (leczy 10/15/25HP). Możemy też użyć obrazka jedzenia z innego poziomu, aby uleczyć 100HP. Mysz

(Catnip*) - może być biała (czas działania - 15s) lub czerwona (czas działania - 30s). Jeżeli chcemy ustawić własny czas używamy atrybutu *Smarts*.

Smarts*: ?? - dowolna wartość dla Catnipa, 1 to ms (milisekunda), więc jeżeli chcemy dodać mysz, która ma działać np. przez 20 sekund, wpisujemy 20000

Przykład grafiki (Poziom1*)	Nazwa grafiki	Domyślna wartość Z
	LEVEL_HEALTH	1000
	GAME_HEALTH_POTION3	
	GAME_CATNIPS_NIP1	5000

2.100 Logika HermitCrab || zwykły krab. Występuje tylko w poziomie VII.

Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny


Ten przeciwnik może mieć przy sobie tylko jeden skarb/powerup.

Możemy również zmienić zwykłego kraba na takiego co rzuca bomby. Wchodzimy w i wpisujemy wartość 1 w **User 1**:

Damage: ?? - wartość obrażenia jakie zada nam krab jeżeli na niego wpadniemy

XMin: ??, **XMax**: ?? - jest to obszar w poziomie dla patrolującego przeciwnika

Zawsze XMin < XMax np. XMin: 1000, XMax: 1500.

Grafika	Nazwa grafiki	Domyślna wartość Z
	LEVEL_HERMITCRAB	0

2.101 Logika HitBurst ||

2.102 Logika HugeSoundTrigger || dźwięk wielo lub jednorazowy, który odtwarzany jest kiedy wejdziemy w obszar logiki (zasięg 255 pikseli).

Animation: wstawiamy nazwę dźwięku z tabeli dźwięków


Smarts: -1 - dźwięk będzie odtwarzany za każdym razem, gdy wejdziemy w jego obszar działania lub zostawiamy 0 - dźwięk odtworzy się tylko raz. Wpisanie wartości większej od zera oznacza ile razy możemy "uruchomić" dźwięk. Logika ta resetowana jest przy każdej śmierci Clawa w zależności od wartości *Smarts*.

Dodatkowo zaznaczamy *Always Active* i *No Draw* w .

Grafika	Nazwa grafiki	Domyślna wartość Z
	GAME_SOUNDICON	9000

2.103 Logika IceBullet || pocisk lodowy, który może poruszać się, ale nie musi (np. animowana dekoracja) i jeżeli wpadnie na kafelek o atrybucie: Ground/Solid/Climb to znika, dodatkowy efekt - poświata

SpeedX: ??, **SpeedY**: ?? - prędkość w poziomie/w pionie oraz kąt pod jakim obiekt zacznie się poruszać względem podłoża. Kąt można odpowiednio dobrać posługując się tabelami dla logiki *SkullCannon*. Działa podobnie jak logika *Bullet*.

Grafika	Nazwa grafiki	Domyślna wartość Z
	GAME_PROJECTILES_ICESWORD	0

2.104 Logika JumpSwitch || jeżeli przeciwnik zbliży się do tej logiki przeskoczy w miejsce oznaczone za pomocą XMin, XMax, YMin, YMax. Przeciwnik może użyć tej logiki nieskończoną ilość razy.

XMin: ??, XMax: ??, YMin: ??, YMax: ?? - w ten obszar wskoczy przeciwnik

Dodatkowo zaznaczamy *No Draw* w .

Grafika	Nazwa grafiki	Domyślna wartość Z
	GAME_SOUNDICON	0

c) Logiki od K do O

2.105 **Logika Katherine** || drugi Boss, poziom IV. Posiada przy sobie zielony klejnot (domyślnie).


Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny

User1 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

User2 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

SpeedX: ??, SpeedY: ?? - współrzędne - w to miejsce poleci klejnot po pokonaniu Bossa. Jeżeli zostawimy puste pola, klejnot będzie traktowany jako dekoracja.

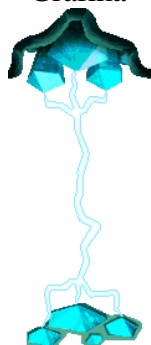
XMin: ??, XMax: ?? - jest to obszar w poziomie dla patrolującego przeciwnika

Grafika	Nazwa grafiki	Domyślna wartość Z
	LEVEL_KATHERINE	0

2.106 **Logika Laser** || laser w poziomie XI

Counter: ?? - odstęp czasowy w ms (milisekundy) między kolejnym wyładowaniem, 1000ms = 1s

Damage: ?? - zadawane obrażenia, domyślna wartość 0 = 1

Grafika	Nazwa grafiki	Domyślna wartość Z
	LEVEL_LASER	0

- 2.107 **Logika LavaGeyser** || gejzer, który rani Clawa. Występuje w poziomie XIV.
Speed: ?? - opóźnienie czasowe. Ustawiamy czas np. na 3s=3000ms, dopiero po tym czasie zacznie działać gejzer.

Grafika



Nazwa grafiki

LEVEL_LAVAGEYSER

Domyślna wartość Z

8005

- 2.108 **Logika LavaHand** || dłoń z lawy rzucająca ognistymi kulami - poziom XIV
Speed: ?? - czas (wartość) w milisekundach pomiędzy kolejnymi wyrzutami kul, 1000ms = 1s
XMin: ??, **XMax:** ??, **YMin:** ??, **YMax:** ?? - obszar aktywowania logiki
Smarts: ?? - liczba rzutów

Grafika



Nazwa grafiki

LEVEL_LAVAHAND

Domyślna wartość Z

8005

- 2.109 Logika LavaHandProjectile || obrazki stanowiące animację dla *LavaHand*

- 2.110 **Logika LavaMouth** || spływająca lava, która rani Clawa w poziomie XIV
Speed: ?? - opóźnienie czasowe. Ustawiamy czas np. na 2s=2000ms, dopiero po tym czasie zacznie spływać lava.

Grafika



Nazwa grafiki

LEVEL_LAVAMOUTH

Domyślna wartość Z

0

- 2.111 **Logika LightningBullet** || pocisk elektryczny, który może poruszać się, ale nie musi (np. animowana dekoracja) i jeżeli wpadnie na kafelek o atrybucie: Ground/Solid/Climb to znika, dodatkowy efekt - poświata
SpeedX: ??, **SpeedY:** ?? - prędkość w poziomie/w pionie oraz kąt pod jakim obiekt zacznie się poruszać względem podłoża. Kąt można odpowiednio dobrać posługując się tabelami dla logiki *SkullCannon*. Działa podobnie jak logika *Bullet*.

Grafika



Nazwa grafiki

GAME_PROJECTILES_LIGHTNINGSWORD

Domyślna wartość
Z

0

2.112 Logika LivesFrame || animacja żyć

2.113 Logika LogicAttack || Attack Rect logiki

2.114 Logika LogicHit || Hit Rect logiki

2.115 **Logika MagicBullet** || pocisk ‘magiczny pazur’, który może poruszać się, ale nie musi (np. animowana dekoracja) i jeżeli wpadnie na kafelek o atrybucie: Ground/Solid/Climb to znika, dodatkowy efekt - poświata
SpeedX: ??, **SpeedY**: ?? - prędkość w poziomie/w pionie oraz kąt pod jakim obiekt zacznie się poruszać względem podłoża. Kąt można odpowiednio dobrać posługując się tabelami dla logiki *SkullCannon*. Działa podobnie jak logika Bullet.



Nazwa grafiki
GAME_MAGICCLAW

Domyślna wartość Z
0

2.116 **Logika MagicPowerup** || amunicja do magicznych pazurów



Nazwa grafiki
GAME_MAGIC_GLOW

Domyślna wartość Z
0

2.117 Logika MapPieceLogic || odpowiedzialna za działanie mappiece

2.118 Logika MapProgressLogic || odpowiedzialna za zdobywanie kolejnych kawałków mapy, animacja po ukończeniu poziomu

2.119 **Logika Marrow** || piąty Boss, poziom X. Posiada przy sobie niebieski klejnot (domyślnie).

Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny

User1 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

User2 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

SpeedX: ??, **SpeedY**: ?? - współrzędne - w to miejsce poleci klejnot po pokonaniu Bossa. Jeżeli zostawimy puste pola, klejnot będzie traktowany jako dekoracja.




Nazwa grafiki
LEVEL_MARROW

Domyślna wartość Z
0

2.120 **Logika MarrowFloor** || rodzaj Elevatora, który działa na określonych zasadach podczas walki z piątym Bossem

SpeedX: ?? - prędkość (wartość) elewatora

XMin: ??, **XMax**: ?? - zasięg wysuwania i wsuwania się Elevatora w ścianę

Grafika	Nazwa grafiki	Domyślna wartość Z
	LEVEL_TRAPELEVATOR	2000/2500

2.121 Logika MenuClaw || odpowiedzialna za animację Clawa w Menu Głównym

2.122 Logika MenuSparkle || odpowiedzialna za animację blasku w Menu Głównym

2.123 Logika Mercat || przeciwnik nr 1 w poziomie XI i XII (inny kolor)

Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny


User1 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

User2 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

Maksymalna ilość skarbów to 9.

XMin: ??, **XMax:** ?? - jest to obszar w poziomie dla patrolującego przeciwnika

Zawsze XMin < XMax np. XMin: 1000, XMax: 1500.

Przykład grafiki (Poziom11)	Nazwa grafiki	Domyślna wartość Z
	LEVEL_MERCAT	0

2.124 Logika Message || wiadomość

2.125 Logika MultiStats || interfejs graficzny (życia, zdrowie, etc.)

2.126 Logika Officer || przeciwnik nr 1 w poziomie I i II

Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny


User1 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

User2 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

Maksymalna ilość skarbów to 9.


XMin: ??, **XMax:** ?? - jest to obszar w poziomie dla patrolującego przeciwnika

Zawsze XMin < XMax np. XMin: 1000, XMax: 1500.


Grafika	Nazwa grafiki	Domyślna wartość Z
	LEVEL_OFFICER	4000

2.127 Logika Omar || ósmy, ostatni Boss, poziom XIV. Posiada przy sobie ostatni klejnot (domyślnie).



SpeedX: ??, SpeedY: ?? - współrzędne - w to miejsce poleci klejnot po pokonaniu Bossa. Jeżeli zostawimy puste pola, klejnot będzie traktowany jako dekoracja.

Grafika	Nazwa grafiki	Domyślna wartość Z
	LEVEL_OMAR	0


- 2.128 **Logika OmarBullet** || wykorzystywana przez inną logikę - Omar. Są to pociski, którymi miota w Clawa ostatni Boss. Może poruszać się, ale nie musi (np. animowana dekoracja) i jeżeli wpadnie na kafelkę o atrybucie: Solid to znika
Damage: ?? - wpisujemy wartość dla zadawanych obrażeń (domyślnie pocisk rani gracza i odbiera mu 1 punkt zdrowia)
SpeedX: ??, SpeedY: ?? - prędkość w poziomie/w pionie oraz kąt pod jakim obiekt zacznie się poruszać względem podłoża. Kąt można odpowiednio dobrać posługując się tabelami dla logiki *SkullCannon*. Działa podobnie jak logika Bullet.

Grafika	Nazwa grafiki	Domyślna wartość Z
	LEVEL_OMARPROJECTILE	0

- 2.129 **Logika OmarShield** || tarcza w postaci krążących wokół Omara ognistych lub lodowych kul
Damage: ?? - wpisujemy wartość dla zadawanych obrażeń (domyślnie kula rani gracza i odbiera mu 1 punkt zdrowia)

Grafika	Nazwa grafiki	Domyślna wartość Z
	LEVEL_FIRESHIELD	0
	LEVEL_ICESHIELD	

- 2.130 **Logika OneTimeStartSteppingStone** || platforma, która zanurza się kiedy Claw na nią wskoczy. Działa jednorazowo.

Przykład grafiki (Poziom4)	Nazwa grafiki	Domyślna wartość Z
	LEVEL_STEPPINGSTONE	4000

- 2.131 **Logika OneWayStartElevator** || platforma, która zaczyna poruszać się dopiero kiedy na nią wskoczymy. Porusza się tylko jeśli na niej stoimy i działa jednorazowo (pokonuje swoją trasę tylko jeden raz). Aby platforma poruszała się w dobrym kierunku trzeba wpisać odpowiednie wartości np. *X* dla platformy wynosi 1000, więc jeśli ma poruszać się w prawo (kierunek: 6) - XMin: 1000, XMax: 1200 <- przykład
XMin: ??, XMax: ?? - zasięg w poziomie (XMin < XMax)
YMin: ??, YMax: ?? - zasięg w pionie (YMin < YMax)
SpeedX: ??, SpeedY: ?? - prędkość w poziomie/w pionie
Direction: ?? - kierunek dla poruszającej się platformy (nie działa*)

7 - lewy górny
4 - w lewo
1 - lewy dolny*

8 - w górę
5 - stoi w miejscu
2 - w dół

9 - prawy górny*
6 - w prawo
3 - prawy dolny

Przykład grafiki (Poziom1)



Nazwa grafiki
LEVEL_ELEVATORS

Domyślna wartość Z
2000

2.132 **Logika OneWayTriggerElevator** || platforma, która zaczyna się poruszać dopiero kiedy na nią wskoczymy, a później już sama się porusza nawet jeśli na niej nie stoimy. Działa jednorazowo (pokonuje swoją trasę tylko jeden raz). Aby platforma poruszała się w dobrym kierunku trzeba wpisać odpowiednie wartości np. Y dla platformy wynosi 2000, więc jeśli ma poruszać się w górę (kierunek: 8) - **YMin**: 1000, **YMax**: 2000 <- przykład.

XMin: ??, **XMax**: ?? - zasięg w poziomie (XMin < XMax)

YMin: ??, **YMax**: ?? - zasięg w pionie (YMin < YMax)

SpeedX: ??, **SpeedY**: ?? - prędkość w poziomie/w pionie

Direction: ?? - kierunek dla poruszającej się platformy (nie działa*)

7 - lewy górny
4 - w lewo
1 - lewy dolny*

8 - w górę
5 - stoi w miejscu
2 - w dół

9 - prawy górny*
6 - w prawo
3 - prawy dolny

Przykład grafiki (Poziom1)



Nazwa grafiki
LEVEL_ELEVATORS

Domyślna wartość Z
2000

d) Logiki od P do S

2.133 **Logika Parrot** || papuga, która atakuje nas podczas walki z piątym Bossem
XMin: ??, **XMax**: ??, **YMin**: ??, **YMax**: ?? - obszar, w którym papuga lata, atakuje Clawa oraz odpoczywa

Grafika



Nazwa grafiki

LEVEL_PARROT

Domyślna wartość Z


0

2.134 **Logika PathElevator** || platforma, która porusza się po wyznaczonej ścieżce, którą ustala się w **Rects...**. Kierunek oznaczamy liczbami od 1 do 9.

<input checked="" type="checkbox"/>	Move Rect
<input checked="" type="checkbox"/>	Left: 0 X Top: 0 Y Right: 0 XBottom: 0 Y
<input checked="" type="checkbox"/>	Hit Rect
<input checked="" type="checkbox"/>	Left: 0 X Top: 0 Y Right: 0 XBottom: 0 Y
<input checked="" type="checkbox"/>	Attack Rect
<input checked="" type="checkbox"/>	Left: 0 X Top: 0 Y Right: 0 XBottom: 0 Y
<input checked="" type="checkbox"/>	Clip Rect
<input checked="" type="checkbox"/>	Left: 0 X Top: 0 Y Right: 0 XBottom: 0 Y

Pola Left: ??, Right: ?? - odpowiadają za kierunek
Pola Top: ??, Bottom: ?? - odpowiadają za odległość, którą platforma ma przebyć
Speed: ?? - wartość prędkości dla platformy
Kierunki:

7 - lewy górny	8 - w górę	9 - prawy górny
4 - w lewo	5 - stoi w miejscu	6 - w prawo
1 - lewy dolny	2 - w dół	3 - prawy dolny

Przykład grafiki (Poziom1)	Nazwa grafiki	Domyślna wartość Z
	LEVEL_ELEVATORS	2000

2.135 **Logika PegLeg** || przeciwnik nr 1 w poziomie IX i X (inny kolor)

Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny


User1 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

User2 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

Maksymalna ilość skarbów to 9.

XMin: ??, **XMax**: ?? - jest to obszar w poziomie dla patrolującego przeciwnika

Zawsze XMin < XMax np. XMin: 1000, XMax: 1500.

Przykład grafiki (Poziom9)	Nazwa grafiki	Domyślna wartość Z
	LEVEL_PEGLEG	5000


2.136 **Logika PistolBullet** || kula z pistoletu, która może poruszać się, ale nie musi (nie posiada animacji) i jeżeli wpadnie na kafelek o atrybucie: Ground/Solid/Climb to znika

SpeedX: ??, **SpeedY**: ?? - prędkość w poziomie/w pionie oraz kąt pod jakim obiekt zacznie się poruszać względem podłoża. Kąt można odpowiednio dobrać posługując się tabelami dla logiki *SkullCannon*. Działa podobnie jak logika Bullet.

Grafika	Nazwa grafiki	Domyślna wartość Z
	GAME_BULLETS	0

2.137 **Logika PointsIcon** || działa jak animacja punktów tzn. pojawia się obiekt, wznosi się na kilkanaście pikseli po czym znika (obiekt nie odtwarza animacji)

2.138 **Logika PowderKeg** || beczka, którą można podnieść i rzucić w przeciwnika

Przykład grafiki (Poziom1)	Nazwa grafiki	Domyślna wartość Z
	LEVEL_POWDERKEG	5000

2.139 **Logika PowerupGlitter** || animacja połysku. Może stanowić element dekoracyjny.

Grafika



Nazwa grafiki

GAME_GLITTER

Domyślna wartość Z

0

2.140 **Logika PunkRat** || szczur siedzący na armacie i strzelający z niej. Logika łączy ze sobą dwa obiekty: PUNKRAT i CANNON. Występuje w kilku poziomach. Nie może mieć przy sobie skarbów.

Przykład grafiki (Poziom2)



Nazwa grafiki

LEVEL_PUNKRAT

Domyślna wartość Z

0

2.141 **Logika Rat** || szczur, który występuje w kilku poziomach. Nie może mieć przy sobie skarbów.

XMin: ??, **XMax:** ?? - jest to obszar w poziomie dla patrolującego przeciwnika

Przykład grafiki (Poziom1)



Nazwa grafiki

LEVEL_RAT

Domyślna wartość Z

0

2.142 **Logika RatBomb** || działa podobnie jak bomba szczura, z tym, że wybucha dopiero po zetknięciu się z kafelkiem: Solid/Ground/Climb. Odbiera 15HP.

SpeedX: ??, **SpeedY:** ?? - prędkość w poziomie/w pionie oraz kąt pod jakim obiekt zacznie się poruszać względem podłoża. Kąt można odpowiednio dobrać posługując się tabelami dla logiki *SkullCannon*.

Grafika



Nazwa grafiki

LEVEL_RATBOMB

Domyślna wartość Z

0

2.143 **Logika Raux** || pierwszy Boss, poziom II. Posiada przy sobie niebieski klejnot (domyślnie).

Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny

User1 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

User2 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

SpeedX: ??, **SpeedY:** ?? - współrzędne - w to miejsce poleci klejnot po pokonaniu Bossa. Jeśli zostawimy puste pola, klejnot będzie traktowany jako dekoracja.

XMin: ??, **XMax:** ?? - jest to obszar w poziomie dla patrolującego przeciwnika

Grafika




Nazwa grafiki

LEVEL_RAUX

Domyślna wartość Z

0


- 2.144 **Logika RedTail** || siódmy Boss, poziom XIII. Posiada przy sobie czerwony i niebieski klejnot (domyślnie)
SpeedX: ??, SpeedY: ?? - współrzędne - w to miejsce poleci klejnot po pokonaniu Bossa. Jeżeli zostawimy puste pola, klejnot będzie traktowany jako dekoracja.

Grafika	Nazwa grafiki	Domyślna wartość Z
	LEVEL_REDTAIL	9000


- 2.145 **Logika RedTailBullet** || pocisk, którego używa przedostatni Boss. Można wykorzystać tę logikę jako jednorazowy pocisk raniący Clawa (dowolna grafika). Jeżeli obiekt nie trafi w Clawa, to zniknie po zderzeniu się z kafelkiem: Solid/Ground/Climb. Jeżeli zderzy się z Clawem, to zostanie już na zawsze w tym miejscu i będzie cały czas ranić Clawa.
SpeedX: ??, SpeedY: ?? - prędkość w poziomie/w pionie oraz kąt pod jakim obiekt zacznie się poruszać względem podłoża. Kąt można odpowiednio dobrać posługując się tabelami dla logiki *SkullCannon*.
Damage: ?? - wartość zadawanych obrażeń

- 2.146 **Logika RedTailKnife** || używana przez siódmego Bossa

- 2.147 **Logika RedTailPirate** || pirat. Występuje w kilku poziomach.
Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny
User1 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów
User2 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów
Maksymalna ilość skarbów to 9.
XMin: ??, XMax: ?? - jest to obszar w poziomie dla patrolującego przeciwnika
Zawsze XMin < XMax np. XMin: 1000, XMax: 1500.

Przykład grafiki (Poziom7)	Nazwa grafiki	Domyślna wartość Z
	LEVEL_REDTAILPIRATE	30

- 2.148 **Logika RedTailSpikes** || kolce na które można wpaść podczas walki z siódmym Bossem. Zadają 20HP obrażeń.

Grafika	Nazwa grafiki	Domyślna wartość Z
	LEVEL_BOSSSPIKES	1000

2.149 **Logika RedTailWind** || wiatr, który spycha Clawa na kolce podczas walki z siódmym Bossem

Dodatkowo zaznaczamy *No Draw* w .

Grafika



Nazwa grafiki

GAME_SOUNDICON

Domyślna wartość

10000

2.150 **Logika RobberThief** || przeciwnik nr 1 w poziomie III i IV (inny kolor)

Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny

User1 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

User2 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

Maksymalna ilość skarbów to 9.

XMin: ??, **XMax:** ?? - jest to obszar w poziomie dla patrolującego przeciwnika

Zawsze XMin < XMax np. XMin: 1000, XMax: 1500.

Przykład grafiki (Poziom3)



Nazwa grafiki

LEVEL_ROBBERTHIEF

Domyślna wartość Z

0

2.151 **Logika SawBlade** || obracająca się piła. Wychodzi z podłoża i rani Clawa.

SpeedX: ?? - czas (wartość w tysiącach) na jaki pojawi się piła w ms (milisekundy), 1000ms = 1s

SpeedY: ?? - czas (wartość w tysiącach) na jaki piła schowa się w ms (milisekundy), 1000ms = 1s

Speed: ?? - opóźnienie względem kolejnych obiektów z logiką SawBlade2/3/4

Grafika



Nazwa grafiki

LEVEL_FLOORSAW

Domyślna wartość Z

0

2.152 **Logika SawBlade2** || obracająca się piła. Wychodzi z podłoża i rani Clawa.

SpeedX: ?? - czas (wartość w tysiącach) na jaki pojawi się piła w ms (milisekundy), 1000ms = 1s

SpeedY: ?? - czas (wartość w tysiącach) na jaki piła schowa się w ms (milisekundy), 1000ms = 1s

Speed: ?? - opóźnienie względem kolejnych obiektów z logiką SawBlade/3/4

Grafika



Nazwa grafiki

LEVEL_FLOORSAW

Domyślna wartość Z

0

2.153 **Logika SawBlade3** || obracająca się piła. Wychodzi z podłoża i rani Clawa.

SpeedX: ?? - czas (wartość w tysiącach) na jaki pojawi się piła w ms (milisekundy), 1000ms = 1s

SpeedY: ?? - czas (wartość w tysiącach) na jaki piła schowa się w ms (milisekundy),
1000ms = 1s
Speed: ?? - opóźnienie względem kolejnych obiektów z logiką SawBlade/2/4



Grafika	Nazwa grafiki	Domyślna wartość Z
	LEVEL_FLOORSAW	0

- 2.154 **Logika SawBlade4** || obracająca się piła. Wychodzi z podłoża i rani Clawa.
SpeedX: ?? - czas (wartość w tysiącach) na jaki pojawi się piła w ms (milisekundy),
1000ms = 1s
SpeedY: ?? - czas (wartość w tysiącach) na jaki piła schowa się w ms (milisekundy),
1000ms = 1s
Speed: ?? - opóźnienie względem kolejnych obiektów z logiką SawBlade/2/3



Grafika	Nazwa grafiki	Domyślna wartość Z
	LEVEL_FLOORSAW	0

2.155 Logika ScoreFrame || animacja punktów

2.156 Logika ScoreRibbon ||

- 2.157 **Logika Seagull** || mewa, występuje w kilku poziomach. Widzi niewidzialnych.
Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny
User1 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów
User2 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów
Maksymalna ilość skarbów to 9.
XMin: ??, **XMax:** ?? - jest to obszar w poziomie dla patrolującego przeciwnika
Zawsze XMin < XMax np. XMin: 1000, XMax: 1500.



Grafika	Nazwa grafiki	Domyślna wartość Z
	LEVEL_SEAGULL	0

2.158 Logika SecondSuperCheckpoint || drugi punkt zapisu



Grafika	Nazwa grafiki	Domyślna wartość Z
	GAME_SUPERCHECKPOINT	1000

- 2.159 Logika Shake || efekt trzęsienia ziemi - drgania ekranu
Attack Rect: ??, ??, ??, ?? - obszar aktywowania logiki

Dodatkowo zaznaczamy *No Draw* w .

Grafika



Nazwa grafiki

GAME_SOUNDICON

Domyślna wartość Z

0

2.160 Logika SimpleAnimation ||

2.161 Logika SingleFrameMessage ||

2.162 Logika SinglePlank ||

2.163 Logika Siren || przeciwnik nr 2 w poziomie XI i XII (inny kolor)

Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny

User1 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

User2 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

Maksymalna ilość skarbów to 9.

XMin: ??, **XMax:** ?? - jest to obszar w poziomie dla patrolującego przeciwnika

Zawsze XMin < XMax np. XMin: 1000, XMax: 1500.

Przykład grafiki (Poziom11)



Nazwa grafiki

LEVEL_SIREN

Domyślna wartość Z

5000

2.164 Logika SirenBullet || używa jej logika Siren. Paraliżuje Clawa na 3 sekundy.

2.165 Logika SkullCannon || dział, które strzela kulami w Clawa

SpeedX: ?? - prędkość pocisku w poziomie (liczba pikseli/sekunda)

SpeedY: ?? - prędkość pocisku w pionie (liczba pikseli/sekunda)

Powyższe pola pozwalają również ustalić kąt nachylenia pod jakim pociski będą lecieć. Domyślna prędkość pocisku = 500 pikseli/s. Tabela poniżej przedstawia kilka wybranych kątów i prędkości. Jeżeli chcemy uzyskać np. 2 razy wyższą prędkość należy pomnożyć właśnie przez 2 wartość w *SpeedX* i w *SpeedY*. Pierwsze dwie tabele przedstawiają wartości dla działu domyślnie strzelającego **w prawo**. Dwie kolejne dla działu strzelającego **w lewo**:


Kąt	Prędkość	SpeedX	SpeedY
15	250	241	64
30	250	216	125
45	250	152	152
60	250	75	130
75	250	19	71
90	250	1	250

Kąt	Prędkość	SpeedX	SpeedY
270	250	1	-250
285	250	64	-241
300	250	124	-216
315	250	176	-176
330	250	216	-125
345	250	241	-64


Kąt	Prędkość	SpeedX	SpeedY
15	250	241	-64
30	250	216	-125
45	250	176	-176
60	250	124	-216
75	250	64	-241
90	250	1	-250

Kąt	Prędkość	SpeedX	SpeedY
270	250	1	250
285	250	64	241
300	250	124	216
315	250	176	176
330	250	216	125
345	250	241	64

Kąty dodawane są do kąta początkowego (0°) zgodnie z ruchem wskazówek zegara. Domyślnie działo skierowane jest w prawo, więc jeżeli pocisk ma lecieć w lewo pod odpowiednim kątem należy zaznaczyć flagę Mirror w .

Grafika	Nazwa grafiki	Domyślna wartość Z
	LEVEL_SKULLCANNON	0

- 2.166 **Logika SlidingElevator** || platforma, która się wysuwa i wsuwa
SpeedX: ?? - czas (wartość) w ms, (1000ms = 1s), można zostawić 0 (domyślnie)

Przykład grafiki (Poziom2)	Nazwa grafiki	Domyślna wartość Z
	LEVEL_PEGSLIDER	2000

- 2.167 **Logika SmallSoundTrigger** || odtwarzanie dźwięku np. otworzenie drzwi lub jednorazowe dźwięki, które są odtwarzane kiedy wejdziemy w obszar logiki (zasięg 30 pikseli)

Animation: wstawiamy nazwę dźwięku z tabeli dźwięków

Smarts: -1 - dźwięk będzie odtwarzany za każdym razem, gdy wejdziemy w jego obszar działania lub zostawiamy 0 - dźwięk odtworzy się tylko raz. Wpisanie wartości większej od zera oznacza ile razy możemy "uruchomić" dźwięk. Logika ta resetowana jest przy każdej śmierci Clawa w zależności od wartości *Smarts*.

Dodatkowo zaznaczamy *Always Active* i *No Draw* w .

Grafika



Nazwa grafiki

GAME_SOUNDICON

Domyślna wartość Z

9000

2.168 **Logika Soldier** || przeciwnik nr 2 w poziomie I i II

Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny

User1 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

User2 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

Maksymalna ilość skarbów to 9.

XMin: ??, **XMax:** ?? - jest to obszar w poziomie dla patrolującego przeciwnika

Zawsze XMin < XMax np. XMin: 1000, XMax: 1500.

Grafika



Nazwa grafiki

LEVEL_SOLDIER

Domyślna wartość Z

4000

2.169 **Logika SoundTrigger** || odtwarzanie dźwięku np. kałuża lub jednorazowe dźwięki, które są odtwarzane kiedy wejdziemy w obszar logiki (zasięg 65 pikseli)

Animation: wstawiamy nazwę dźwięku z tabeli dźwięków

Smarts: -1 - dźwięk będzie odtwarzany za każdym razem, gdy wejdziemy w jego obszar działania lub zostawiamy 0 - dźwięk odtworzy się tylko raz. Wpisanie wartości większej od zera oznacza ile razy możemy "uruchomić" dźwięk. Logika ta resetowana jest przy każdej śmierci Clawa w zależności od wartości *Smarts*.

Damage: ?? - głośność dźwięku np. 150

Dodatkowo zaznaczamy *Always Active* i *No Draw* w .

Grafika



Nazwa grafiki

GAME_SOUNDICON

Domyślna wartość Z

9000

2.170 **Logika SpecialPowerup** || dodatkowe życie, niewidzialność*, nietykalność*, lodowy miecz*, ognisty miecz*, elektryczny miecz*, mappiece, flaga (koniec poziomu), klejnot (koniec poziomu), teleport**

Smarts*: ?? - czas działania Powerupa, w ms - 1000ms = 1s

Damage: 0/1 - nie odnawiający/odnawiający się miecz ognisty/lodowy/elektryczny

SpeedX:** ??, **SpeedY**:** ?? - współrzędne, w które zostaniemy przeniesieni, dodatkowo możemy ustawić *Smarts:* 0/1 - działa raz/działa nieskończenie wiele razy

Grafika



Nazwa grafiki

GAME_POWERUPS_EXTRALIFE

Domyślna wartość Z

1000

GAME_POWERUPS_INVULNERABLE

0

GAME_WARP

0

2.171 **Logika Splash** || jednorazowa animacja Splash (efekt dekoracyjny), po prostu *Splash*

2.172 **Logika SpotAmbientSound** || odtwarzanie dźwięku np. łańcuchów/roślinności itp., który jest odtwarzany cały czas, a obszar działania logiki to ~635 pikseli

Animation: wstawiamy nazwę dźwięku z tabeli dźwięków

Dodatkowo zaznaczamy *Always Active* i *No Draw* w **Flags...**.

Grafika



Nazwa grafiki

GAME_SOUNDICON

Domyślna wartość Z

9000

2.173 **Logika SpringBoard** || paprotka/fontanna. Wystrzeliwuje Clawa do góry.

YMax: ?? - 0 wartość domyślna = 450. Można również wpisać własną wartość np. 300 tzn. Claw wyleci 300 pikseli do góry.

Przykład grafiki



Nazwa grafiki

LEVEL_SPRINGYFERN

Domyślna wartość Z

2000

2.174 **Logika Stalactite** || stalaktyt, który spada i rani Clawa

User1 Rect: ??, ??, ??, ?? - obszar aktywowania logiki

Grafika



Nazwa grafiki

LEVEL_PROJECTILES_STALACTITE

Domyślna wartość Z

1010

2.175 **Logika StandardElevator** || działa nawet bez wpisywania wartości w pola. Porusza się z domyślną prędkością 125 pikseli/s i ma zasięg 200 pikseli w prawo, później wraca - 200 pikseli w lewo i tak w kółko. Wstawiając kilka platform blisko siebie i wpisując różne wartości dla prędkości możemy zauważyć, że platformy nie zawsze będą uruchamiać się w tym samym momencie.

XMin: ?? - odległość w pikselach, wybierając tylko to pole platforma porusza się w poziomie

XMax: ?? - odległość w pikselach, wybierając tylko to pole platforma porusza się najpierw w kierunku 7 (lewy górny) - wpisana wartość, a później w dół (wpisana wartość), później do góry (wpisana wartość) i w dół itd.

XMin: ??, **XMax:** ?? - taka sama wartość - poruszanie się w kierunku 7-3 (lewy górny-prawy dolny)

- różna wartość - ścieżka równoległoboku

YMin: ?? - wartość minimalna Y np. 2000 przy Y obiektu = 2500. Wybierając tylko to pole platforma porusza się po ścieżce równoległoboku. Im większa różnica tym bardziej "pionowa" ścieżka równoległoboku.

YMax: ?? - wartość maksymalna Y np. 2800 przy Y obiektu = 2600. Wybierając tylko to pole platforma porusza się po ścieżce równoległoboku. Im większa różnica tym bardziej "pionowa" ścieżka równoległoboku.

YMin: ??, **YMax:** ?? - taka sama wartość/różna wartość - platforma porusza się po ścieżce równoległoboku

SpeedX: ??, **SpeedY:** ?? -a) X = 200, Y = 500 - ścieżka równoległoboku "pozioma"

-b) X = 500, Y = 200 - ścieżka równoległoboku "pionowa"

-c) X,Y = 300 - platforma porusza się w kierunku 7-3

SpeedX: ??, **SpeedY:** ?? - a) b) c) (wartości jak powyżej) - ścieżka równoległoboku "pionowa". Ponadto im wyższa wartość *SpeedY* tym bardziej zwężona ścieżka przy zachowaniu małej wartości *SpeedX* np. 200 lub dużej różnicy pomiędzy *SpeedY*, a *SpeedX* np. 300.

Przykład grafiki (Poziom1)



Nazwa grafiki

LEVEL_ELEVATORS

Domyślna wartość Z

2000

2.176 **Logika StartElevator** || platforma, która zaczyna się poruszać dopiero kiedy na nią wskoczymy. Porusza się tylko jeśli na niej stoimy.

XMin: ??, **XMax:** ?? - zasięg w poziomie (XMin < XMax)

YMin: ??, **YMax:** ?? - zasięg w pionie (YMin < YMax)

SpeedX: ??, **SpeedY:** ?? - prędkość w poziomie/w pionie

Przykład grafiki (Poziom1)



Nazwa grafiki

LEVEL_ELEVATORS

Domyślna wartość Z

2000

2.177 **Logika StartSteppingStone** || platforma, która zanurza się kiedy Claw na nią wskoczy. Po 2 sekundach znów się pojawia.

Przykład grafiki (Poziom4)



Nazwa grafiki

LEVEL_STEPPINGSTONE

Domyślna wartość Z

4000

2.178 **Logika StationaryLight** || obiekt znika po 2 sekundach

2.179 **Logika StatusNumberDigit** ||

2.180 **Logika SteppingStone** || platforma, która zanurza się kiedy Claw na nią wskoczy. Możemy ustalić na jak długo się zanurzy i wypłynie.

SpeedX: ??, **SpeedY:** ?? - wypłynięcie i zanurzenie się platformy, czas podajemy w ms (milisekundy), 1000ms = 1s

Przykład grafiki (Poziom4)



Nazwa grafiki

LEVEL_STEPPINGSTONE

Domyślna wartość Z

2000

2.181 **Logika SteppingStone2** || platforma, która zanurza się kiedy Claw na nią wskoczy. Możemy ustalić na jak długo się zanurzy i wypłynie.

SpeedX: ??, **SpeedY:** ?? - wypłynięcie i zanurzenie się platformy, czas podajemy w ms (milisekundy), 1000ms = 1s

Przykład grafiki (Poziom4)



Nazwa grafiki
LEVEL_STEPPINGSTONE

Domyślna wartość Z
2000

- 2.182 **Logika SteppingStone3** || platforma, która zanurza się kiedy Claw na nią wskoczy. Możemy ustalić na jak długo się zanurzy i wypłynie.
SpeedX: ??, **SpeedY**: ?? - wypłynięcie i zanurzenie się platformy, czas podajemy w ms (milisekundy), 1000ms = 1s

Przykład grafiki (Poziom4)



Nazwa grafiki
LEVEL_STEPPINGSTONE

Domyślna wartość Z
2000

- 2.183 **Logika SteppingStone4** || platforma, która zanurza się kiedy Claw na nią wskoczy. Możemy ustalić na jak długo się zanurzy i wypłynie.
SpeedX: ??, **SpeedY**: ?? - wypłynięcie i zanurzenie się platformy, czas podajemy w ms (milisekundy), 1000ms = 1s

Przykład grafiki (Poziom4)



Nazwa grafiki
LEVEL_STEPPINGSTONE

Domyślna wartość Z
2000

Twórcy zadbali o to, aby łatwiej wstawiało się kolejne takie platformy. Wystarczy wstawić je zgodnie z numeracją.

- 2.184 **Logika StopElevator** || platforma, która porusza się. Kiedy na nią wskoczymy, zatrzymuje się. Zeskoczenie z niej ponownie wprawia w ruch platformę.
XMin: ??, **XMax**: ?? - zasięg w poziomie ($XMin < XMax$)
YMin: ??, **YMax**: ?? - zasięg w pionie ($YMin < YMax$)
SpeedX: ??, **SpeedY**: ?? - prędkość w poziomie/w pionie

Przykład grafiki (Poziom1)



Nazwa grafiki
LEVEL_ELEVATORS

Domyślna wartość Z
2000

- 2.185 Logika SuperCheckpointAttack || Rect dla punktów zapisu 1 i 2

- 2.186 Logika SuperPowerup ||

e) Logiki od T do Z

- 2.187 **Logika TallSoundTrigger** || odtwarzanie dźwięku wiele lub jednorazowo, który jest odtwarzany kiedy wejdziemy w obszar logiki (zasięg 25 pikseli w poziomie i 95 pikseli w pionie)
Animation: wstawiamy nazwę dźwięku z tabeli dźwięków
Smarts: -1 - dźwięk będzie odtwarzany za każdym razem, gdy wejdziemy w jego obszar działania lub zostawiamy 0 - dźwięk odtworzy się tylko raz. Wpisanie wartości większej od zera oznacza ile razy możemy "uruchomić" dźwięk. Logika ta resetowana jest przy każdej śmierci Clawa w zależności od wartości *Smarts*.
Dodatkowo zaznaczamy *Always Active* i *No Draw* w .

Grafika



Nazwa grafiki

GAME_SOUNDICON

Domyślna wartość Z

9000

2.188 **Logika Tentacle** || macka, która rani lub chwytą i rani Clawa

Grafika



Nazwa grafiki

LEVEL_AQUATISTENTACLE

Domyślna wartość Z

0

2.189 **Logika TigerGuard** || tygrys. Występuje w poziomie XIV. Istnieją dwie odmiany.

Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny

User1 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

User2 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

Maksymalna ilość skarbów to 9.

XMin: ??, **XMax:** ?? - jest to obszar w poziomie dla patrolującego przeciwnika

Zawsze **XMin** < **XMax** np. XMin: 1000, XMax: 1500.

Możemy również zamienić pomarańczowego tygrysa na białego wpisując w *Smarts* wartość 1.

Przykład grafiki



Nazwa grafiki

LEVEL_TIGER

Domyślna wartość Z

0

2.190 **Logika TimerFrame** || animacji czasu/zegarka2.191 **Logika TinySoundTrigger** || odtwarzanie dźwięku wiele lub jednorazowo, który jest odtwarzany kiedy wejdziemy w obszar logiki (zasięg 10 pikseli)

Animation: wstawiamy nazwę dźwięku z tabeli dźwięków

Smarts: -1 - dźwięk będzie odtwarzany za każdym razem, gdy wejdziemy w jego obszar działania lub zostawiamy 0 - dźwięk odtworzy się tylko raz. Wpisanie wartości większej od zera oznacza ile razy możemy "uruchomić" dźwięk. Logika ta resetowana jest przy każdej śmierci Clawa w zależności od wartości *Smarts*.

Dodatkowo zaznaczamy *Always Active* i *No Draw* w **Flags...**

Grafika



Nazwa grafiki

GAME_SOUNDICON

Domyślna wartość Z

9000

2.192 **Logika TogglePeg** || pojawiająca się i znikająca platforma. Domyślnie nie trzeba nic wpisywać w pola tego obiektu.

SpeedX: ?? - czas (wartość w tysiącach) na jaki pojawi się platforma w ms (milisekundy), 1000ms = 1s

SpeedY: ?? - czas (wartość w tysiącach) na jaki platforma zniknie w ms (milisekundy), 1000ms = 1s

Speed: ?? - opóźnienie względem kolejnych obiektów z logiką TogglePeg2/3/4

Przykład grafiki (Poziom1)



Nazwa grafiki

LEVEL_PEG

Domyślna wartość Z

0

2.193 **Logika TogglePeg2** || pojawiająca się i znikająca platforma. Domyślnie nie trzeba nic wpisywać w pola tego obiektu.

SpeedX: ?? - czas (wartość w tysiącach) na jaki pojawi się platforma w ms (milisekundy), 1000ms = 1s

SpeedY: ?? - czas (wartość w tysiącach) na jaki platforma zniknie w ms (milisekundy), 1000ms = 1s

Speed: ?? - opóźnienie względem kolejnych obiektów z logiką TogglePeg/3/4

Przykład grafiki (Poziom1)



Nazwa grafiki

LEVEL_PEG

Domyślna wartość Z

0

2.194 **Logika TogglePeg3** || pojawiająca się i znikająca platforma. Domyślnie nie trzeba nic wpisywać w pola tego obiektu.

SpeedX: ?? - czas (wartość w tysiącach) na jaki pojawi się platforma w ms (milisekundy), 1000ms = 1s

SpeedY: ?? - czas (wartość w tysiącach) na jaki platforma zniknie w ms (milisekundy), 1000ms = 1s

Speed: ?? - opóźnienie względem kolejnych obiektów z logiką TogglePeg/2/4

Przykład grafiki (Poziom1)



Nazwa grafiki

LEVEL_PEG

Domyślna wartość Z

0

2.195 **Logika TogglePeg4** || pojawiająca się i znikająca platforma. Domyślnie nie trzeba nic wpisywać w pola tego obiektu.

SpeedX: ?? - czas (wartość w tysiącach) na jaki pojawi się platforma w ms (milisekundy), 1000ms = 1s

SpeedY: ?? - czas (wartość w tysiącach) na jaki platforma zniknie w ms (milisekundy), 1000ms = 1s

Speed: ?? - opóźnienie względem kolejnych obiektów z logiką TogglePeg/2/3

Przykład grafiki (Poziom1)



Nazwa grafiki

LEVEL_PEG

Domyślna wartość Z

0

Twórcy gry zadbali o to, aby cztery powyższe logiki współgrały ze sobą. To znaczy, że platformy będą się pojawiać i znikać po kolei. Jeżeli chcemy, aby te platformy pojawiały się na taki sam czas i na taki sam znikwały (nasz ustalony czas), musimy wpisać te same wartości w *SpeedX* i *SpeedY*.

- 2.196 **Logika TowerCannonLeft** || działo, które strzela z wieży w lewą stronę. Zadaje 10 punktów obrażeń.

SpeedX: ??, **SpeedY**: ?? - prędkość w poziomie/w pionie oraz kąt pod jakim obiekt zacznie się poruszać względem podłoża. Kąt można odpowiednio dobrać posługując się tabelami dla logiki *SkullCannon*.

Grafika



Nazwa grafiki

Domyślna wartość Z

LEVEL_TOWERCANNONLEFT

0

- 2.197 **Logika TowerCannonRight** || działo, które strzela z wieży w prawą stronę. Zadaje 10 punktów obrażeń.

SpeedX: ??, **SpeedY**: ?? - prędkość w poziomie/w pionie oraz kąt pod jakim obiekt zacznie się poruszać względem podłoża. Kąt można odpowiednio dobrać posługując się tabelami dla logiki *SkullCannon*.

Grafika



Nazwa grafiki

Domyślna wartość Z

LEVEL_TOWERCANNONRIGHT

0

- 2.198 **Logika TownGuard1** || przeciwnik nr 1 w poziomie V i VI

Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny

User1 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

User2 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

Maksymalna ilość skarbów to 9.

XMin: ??, **XMax**: ?? - jest to obszar w poziomie dla patrolującego przeciwnika

Zawsze XMin < XMax np. XMin: 1000, XMax: 1500.

Grafika



Nazwa grafiki

Domyślna wartość Z

LEVEL_TOWNGUARD1

0

- 2.199 **Logika TownGuard2** || przeciwnik nr 2 w poziomie V i VI

Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny

User1 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

User2 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

Maksymalna ilość skarbów to 9.

XMin: ??, XMax: ?? - jest to obszar w poziomie dla patrolującego przeciwnika
Zawsze XMin < XMax np. XMin: 1000, XMax: 1500.

Grafika

Nazwa grafiki

Domyślna wartość Z



LEVEL_TOWNGUARD2

0

2.200 Logika TProjectile || wylatujące strzały np. poziom IX

Animation: ?? - animacja dla pocisku. Nazwa taka sama jak w polu *Image Set*.

XMin: ??, XMax: ??, YMin: ??, YMax: ?? - obszar, w którym musi znaleźć się Claw, aby aktywować logikę

Direction: ?? - kierunek wystrzelenia pocisku

2: w dół, 4: w lewo, 6: w prawo, 8: w górę

SpeedX: ??, SpeedY: ?? - prędkość pocisku pozioma/pionowa w ms (milisekundy), 1000ms = 1s

Speed: ?? - odstęp czasu pomiędzy wystrzeleniem kolejnych pocisków

W w **User 1: ??** - wpisujemy wartość dla odpowiedniego kierunku. Musi się zgadzać z tym samym kierunkiem, który wybraliśmy w *Direction*. Wartości:

1: w prawo, 2: w lewo, 3: w dół, 4: w górę

Przykład grafiki
(Poziom9)



Nazwa grafiki

Domyślna wartość Z

LEVEL_SHOOTERS_PUFFDARTRIGHT

0

2.201 Logika TreasureCounter ||

2.202 Logika TreasureLogic ||

2.203 Logika TreasurePowerup || skarby. Mają animację GLITTER (połysk). Przy berłach może być wykorzystana flaga Mirror () - odbicie lustrzane.

Przykład grafiki



Nazwa grafiki

Domyślna wartość Z

GAME_TREASURE_SCEPTERS_GREEN

1000

2.204 Logika TridentBullet || pocisk 'trójzab' w poziomie XI. Może się poruszać. Jeżeli napotka na swojej drodze kafelek o atrybucie Ground/Solid/Climb to zniknie. Jeżeli napotka Clawa to zada mu obrażenia i również zniknie. Dodatkowy efekt - poświata. **SpeedX: ??, SpeedY: ??** - prędkość w poziomie/w pionie oraz kąt pod jakim obiekt zacznie się poruszać względem podłoża. Kąt można odpowiednio dobrać posługując się tabelami dla logiki *SkullCannon*. Działa podobnie jak logika Bullet. **Damage: ??** - wartość zadawanych obrażeń Grafiki można użyć dowolnej lub tej z poziomu XI.

- 2.205 **Logika TriggerElevator** || platforma, która zaczyna się poruszać dopiero kiedy na nią wskoczymy. Później już sama się porusza.
XMin: ??, **XMax**: ?? - zasięg w poziomie ($XMin < XMax$)
YMin: ??, **YMax**: ?? - zasięg w pionie ($YMin < YMax$)
SpeedX: ??, **SpeedY**: ?? - prędkość w poziomie/w pionie

Przykład grafiki (Poziom1)



Nazwa grafiki
LEVEL_ELEVATORS

Domyślna wartość Z
 2000

- 2.206 **Logika WaterRock** || fontanna. Wystrzeliwuje Clawa do góry.
YMax: ?? - 0 wartość domyślna = 450, można również wpisać własną wartość np. 300 tzn. Claw wyleci 300 pikseli do góry.

Grafika



Nazwa grafiki

LEVEL_WATERROCK

Domyślna wartość Z

0

- 2.207 Logika WeaponFrame || animacja broni

- 2.208 **Logika WideSoundTrigger** || odtwarzanie dźwięku wiele lub jednorazowo, który jest odtwarzany kiedy wejdziemy w obszar logiki (zasięg 95 pikseli w poziomie i 25 pikseli w pionie)

Animation: wstawiamy nazwę dźwięku z tabeli dźwięków

Smarts: -1 - dźwięk będzie odtwarzany za każdym razem, gdy wejdziemy w jego obszar działania lub zostawiamy 0 - dźwięk odtworzy się tylko raz. Wpisanie wartości większej od zera oznacza ile razy możemy "uruchomić" dźwięk. Logika ta resetowana jest przy każdej śmierci Clawa w zależności od wartości *Smarts*.

Dodatkowo zaznaczamy *Always Active* i *No Draw* w **Flags...**

Grafika



Nazwa grafiki

GAME_SOUNDICON

Domyślna wartość Z

9000

- 2.209 Logika WindDebris ||

- 2.210 **Logika Wolvington** || trzeci Boss, poziom VI. Posiada przy sobie niebieski klejnot (domyślnie).

Powerup: ?? - wpisujemy wartość (liczbę), która odpowiada za skarb (tabela skarbów), dla Catnipa itp. czas zawsze ustawiony jest na domyślny

User1 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

User2 Rect: ??, ??, ??, ?? - wpisujemy jak powyżej wartości dla skarbów

SpeedX: ??, **SpeedY**: ?? - współrzędne - w to miejsce poleci klejnot po pokonaniu Bossa. Jeżeli zostawimy puste pola, klejnot będzie traktowany jako dekoracja.

XMin: ??, **XMax**: ?? - jest to obszar w poziomie dla patrolującego przeciwnika

Grafika

Nazwa grafiki

Domyślna wartość Z



LEVEL_WOLVINGTON

0

2.211 **Logika WolvingtonBullet** || pocisk, którym strzela trzeci Boss. Może poruszać się lub nie. Znika po zetknięciu się z kafelkiem o atrybucie Ground/Solid/Climb, a także kiedy zada obrażenia Clawowi. Dodatkowy efekt - poświata.

SpeedX: ??, **SpeedY**: ?? - prędkość w poziomie/w pionie oraz kąt pod jakim obiekt zacznie się poruszać względem podłoża. Kąt można odpowiednio dobrać posługując się tabelami dla logiki *SkullCannon*. Działa podobnie jak logika *Bullet*.

Damage: ?? - wartość zadawanych obrażeń

Grafika może być dowolna lub ta z poziomu VI.

2.212 **Logika WolvingtonLFX** || poświata dla pocisku wystrzelonego przez Bossa w poziomie VI. Posiada swoją domyślną grafikę.

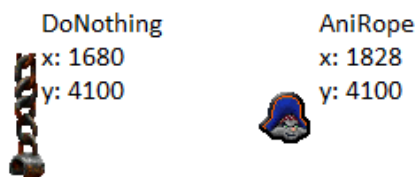
f) Niestandardowe zastosowanie logik

Niestandardowe zastosowanie logik - jest to dodatkowa opcja, której możemy użyć przy tworzeniu poziomu. Warto wiedzieć, że nietypowe rozwiązania wymagają poświęcenia większej uwagi, dlatego należy traktować ostatni podpunkt tej dokumentacji jako element dodatkowy, a nie konieczny.

Logika **AniCycle/AniCycleNormal** - możemy wstawić dowolny obiekt, który może być animowany, a ta animacja może być odtworzona raz lub nieskończoną ilość razy z domyślną lub wybraną prędkością (należy wpisać odpowiednią nazwę w polu *Animation*:). Poniżej został przedstawiony opis animacji jakich możemy użyć (inne odnoszą się do konkretnych obiektów i ich grafik):

Nazwa	Opis
GAME_BACKWARD50	Animacja odtworzy się raz od tyłu z prędkością 0,05s
GAME_BACKWARD100	Animacja odtworzy się raz od tyłu z prędkością 0,1s
GAME_CYCLE50	Animacja odtwarza się w nieskończoność z prędkością 0,05s
GAME_CYCLE100	Animacja odtwarza się w nieskończoność z prędkością 0,1s
GAME_CYCLE200	Animacja odtwarza się w nieskończoność z prędkością 0,2s
GAME_CYCLE500	Animacja odtwarza się w nieskończoność z prędkością 0,5s
GAME_FORWARD50	Animacja odtworzy się raz normalnie z prędkością 0,05s
GAME_FORWARD100	Animacja odtworzy się raz normalnie z prędkością 0,1s
GAME_REVERSECYCLE50	Ani odtwarza się w nieskończoność od tyłu z prędkością 0,05s
GAME_REVERSECYCLE100	Ani odtwarza się w nieskończoność od tyłu z prędkością 0,1s
GAME_REVERSECYCLE200	Ani odtwarza się w nieskończoność od tyłu z prędkością 0,2s
GAME_REVERSECYCLE500	Ani odtwarza się w nieskończoność od tyłu z prędkością 0,5s

Logika **AniRope** - możemy wstawić obiekt tak, aby Claw się go łapał jak liny, a to daje kilka ciekawych rozwiązań. Grafiki lin są tylko w kilku poziomach, więc w innych stosuje się rozwiązanie nie poruszającej się liny. Wstawiając tę logikę w innym poziomie trudno jest ocenić, gdzie wstawić obiekt, aby Claw się go złapał. Poniżej został przedstawiony przykład, jak wykorzystać to rozwiązanie w swoim poziomie.



Po lewej stronie został wstawiony obiekt, którego Claw będzie się chwytać podczas gry. Ma on wpisaną logikę *DoNothing*, ale mógłby mieć również wpisaną *AniCycle*. Po prawej stronie znajduje się obiekt z wpisaną logiką *AniRope* - lina zaczyna swoją animację od strony lewej, dlatego też takie jest rozmieszczenie. Obiekt z tą logiką może mieć dowolną grafikę ponieważ i tak zostanie zaznaczona flaga **NoDraw** w **Flags...**. W praktyce będzie wyglądać to tak:



Należy wpisać w SpeedX wartość 99 999, aby lina nie poruszała się. Unikniemy w ten sposób sytuacji, by nie trzeba było chwytać się niewidzialnej liny w innym miejscu.

Możemy używać logik przeciwników w innych poziomach niż domyślnie oni występują, jednak nie będziemy mogli użyć ich grafik. Obiekty takie nie będą animowane.

Logika **ConveyorBelt** - również tej logiki możemy używać w innym poziomie bez przeszkód. Ze względu na to, że tylko w levelu IX jest grafika dla tego obiektu, w innych używa się niewidzialnych taśm.

Logika **DoNothing/DoNothingNormal** - możemy używać dostępnych klatek animacji danego obiektu, a nawet grafik, które są częścią danej warstwy.

Przykład pierwszy - użycie wybranej klatki animacji przeciwnika jako dodatkowej dekoracji:



Jest to klatka animacji o numerze 104. Aby taka dekoracja pojawiła się w grze należy wpisać liczbę 104 w pole **I: ??**, przy edycji obiektu (obok współrzędnych X,Y,Z). Dostępne numery klatek animacji możemy sprawdzić w folderze **LEVELX → IMAGES → NAZWAPRZECIWNIKA**.

Przykład drugi - wstawienie grafiki z warstwy ACTION (można również z BACK lub FRONT):



Wstawione zostały 4 grafiki z warstwy ACTION. Numery grafik możemy sprawdzić bezpośrednio w edytorze co ułatwia nam zadanie.

W pierwszym przykładzie ImageSet został wypełniony poprzez wpisanie LEVEL_OFFICER, a w drugim przykładzie ACTION.

Elevatory mogą również pełnić funkcję dekoracji np. w poziomie VIII. Przykład poniżej:



W tym przypadku 14 klatka animacji beczki pełni rolę dekoracji jako beczka pływająca w wodzie.

Logika **SpringBoard/WaterRock** - możemy użyć wyrzutni w innych poziomach używając animacji SPLASH dla obiektu lub stworzyć niewidzialną wyrzutnię.