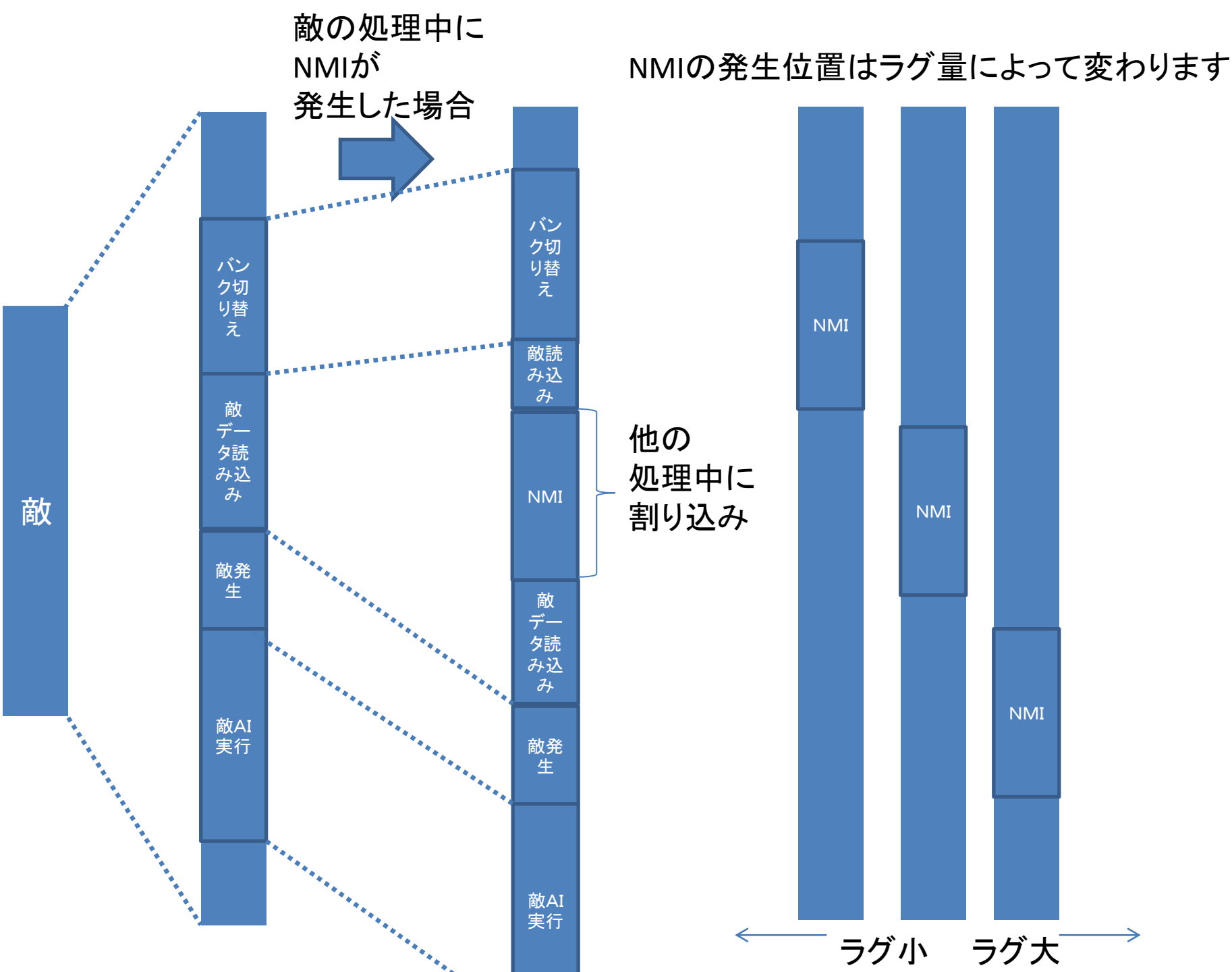
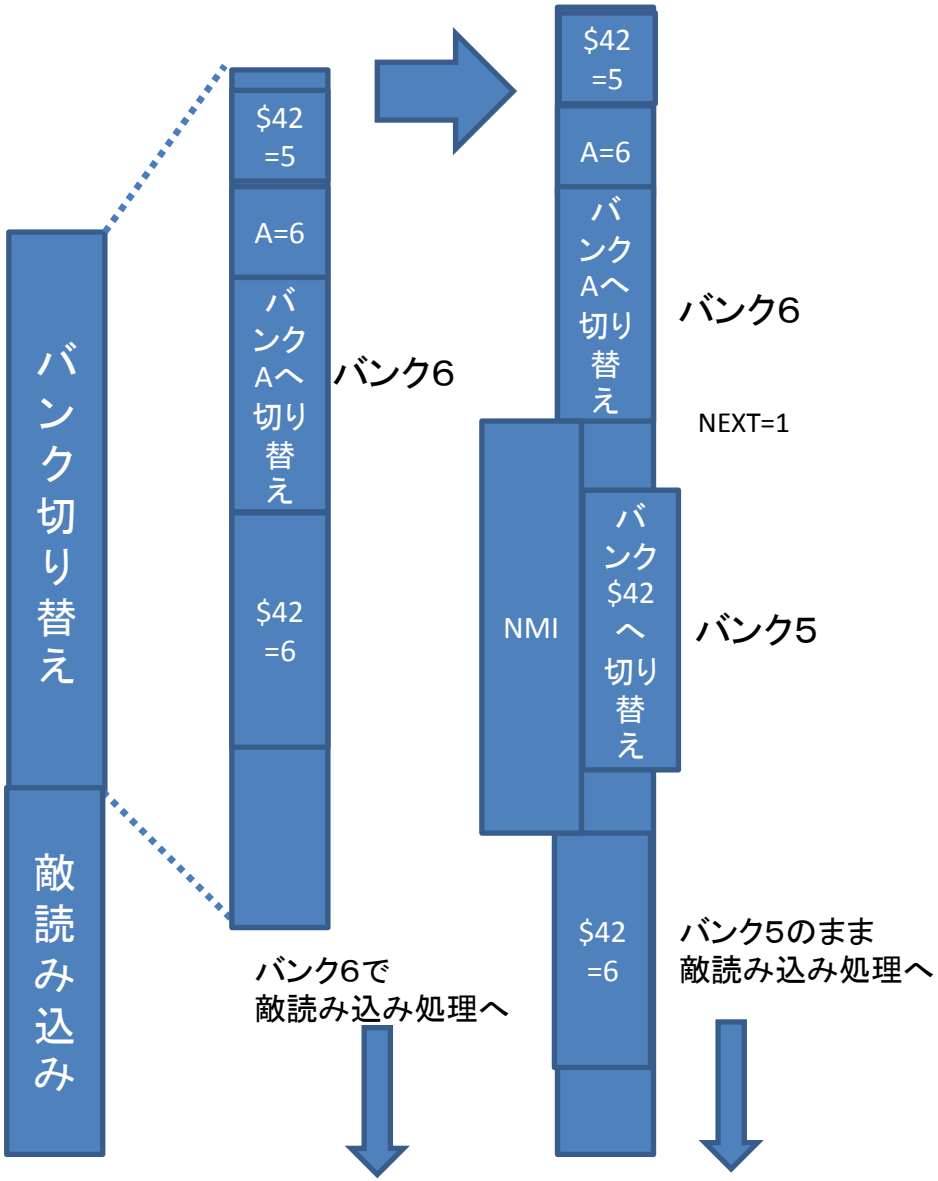
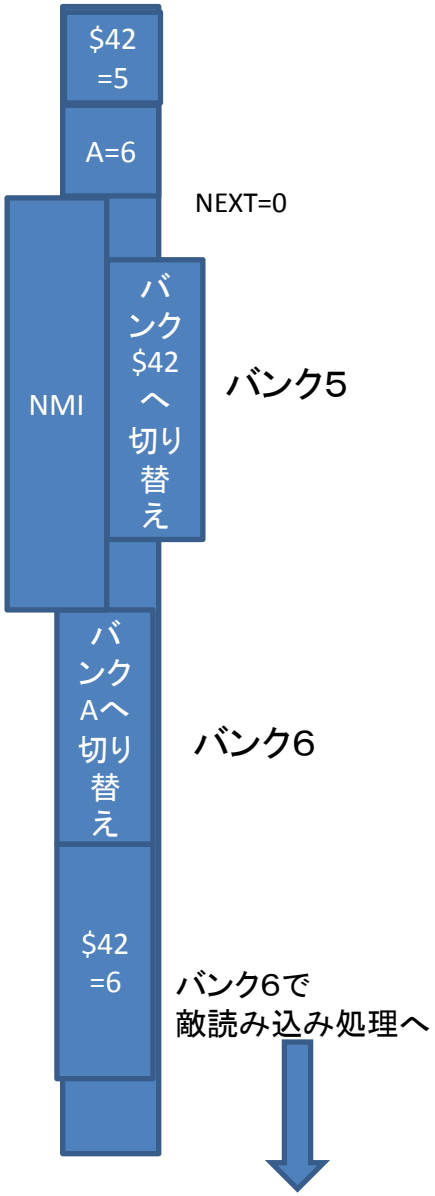
1秒＝60フレーム
1フレーム
1フレーム≒10000命令
NMI
アイテム
ロックマン
敵
NMI
待ち処理
NMIは約10000命令起きに自動的に発生
敵、
アイテムを
増やして
処理数を
増やす
1フレームに処理が収まらなくなり、他の処理中にNMIの処理が発生

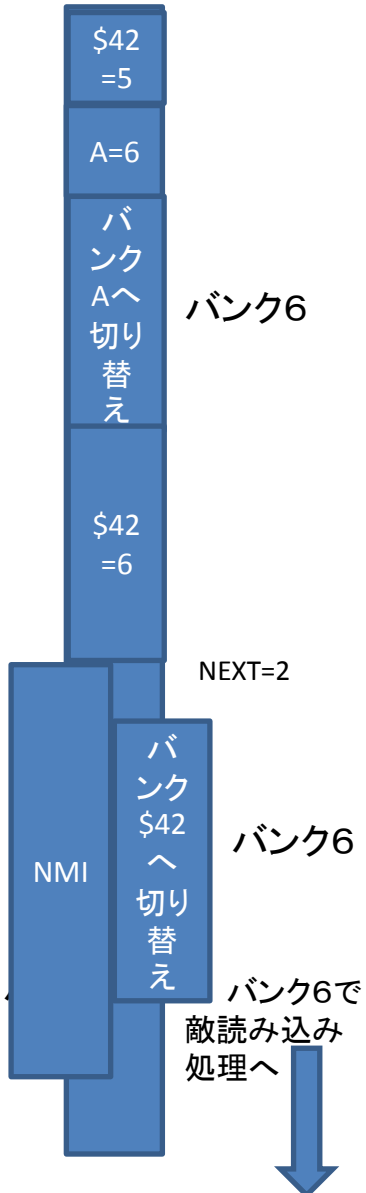

敵の処理中に
NMIが
発生した場合
NMIの発生位置はラグ量によって変わります
敵
バンク切り替え
敵データ読み込み
敵発生
敵AI実行
バンク切り替え
敵読み込み
NMI
敵データ読み込み
敵発生
敵AI実行
他の
処理中に
割り込み
NMI
NMI
NMI
ラグ小
ラグ大

少しでもNMIの位置がずれていると、
この問題は起きません

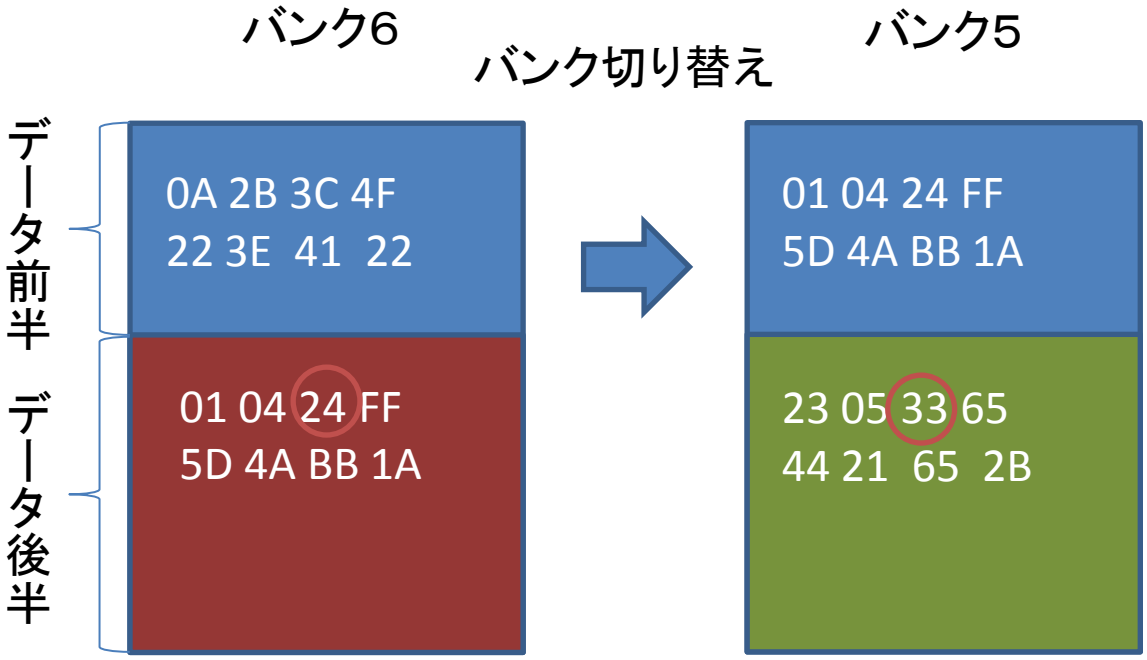
非常に良い位置でNMI発生







なぜバンク5のデータを読み込むと間違いが起きるのか？



本来は「24」というデータを読み込むはずが、
バンクが間違っていることにより「33」を読み込んだりする！

バンク切り替えとは、
アドレス空間の後半のデータを瞬時に切り替えられる機能のこと。
ファミコンではこの機能によって、データ容量不足を防いでいる。

つまりロックマンにおける「ディレイ」の意味は

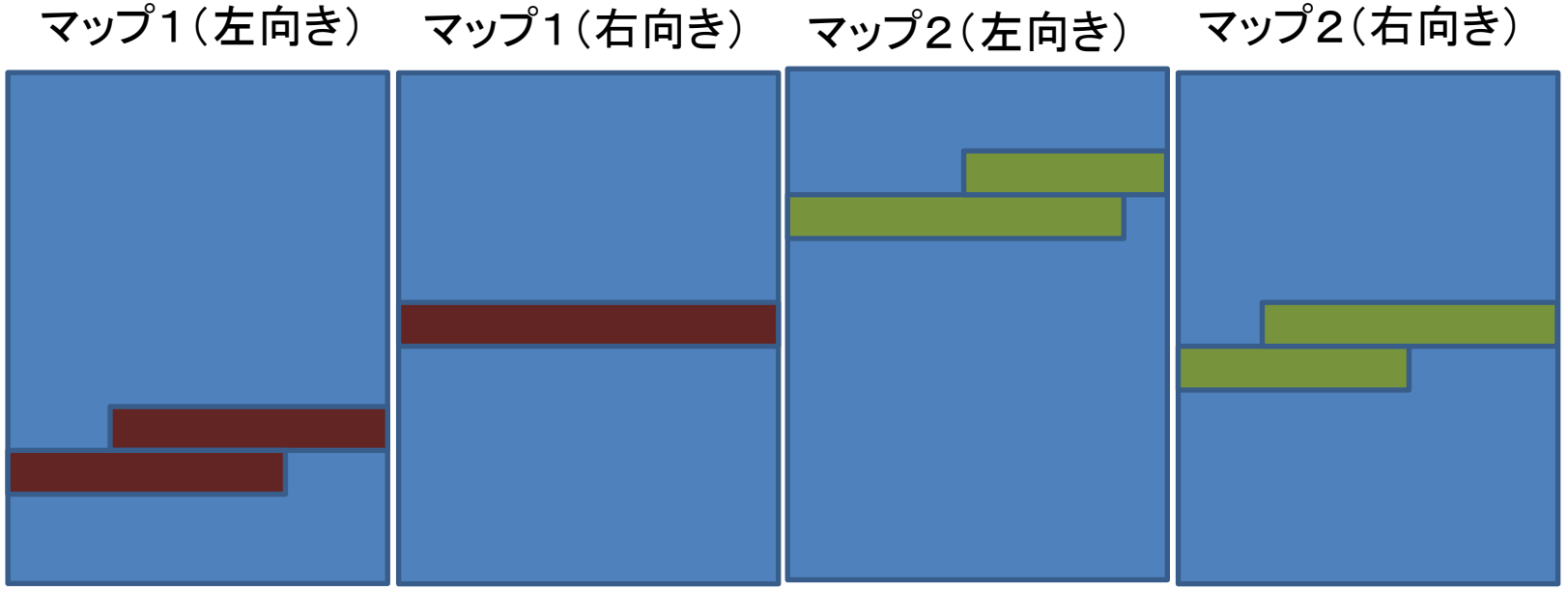
敵やオブジェクトを増加させて
処理量を絶妙に調整することで、NMIを割り込ませると問題が起きる処理に
NMIを割り込ませて読み込ませるデータを狂わせることをあらわす。

（つまり敵やオブジェクトを増やすことによって、
　メモリをあふれさせているわけではなく、オーバーフローとは違うのである）

なぜステージによって呼び出せる敵の種類が決まっているのか？

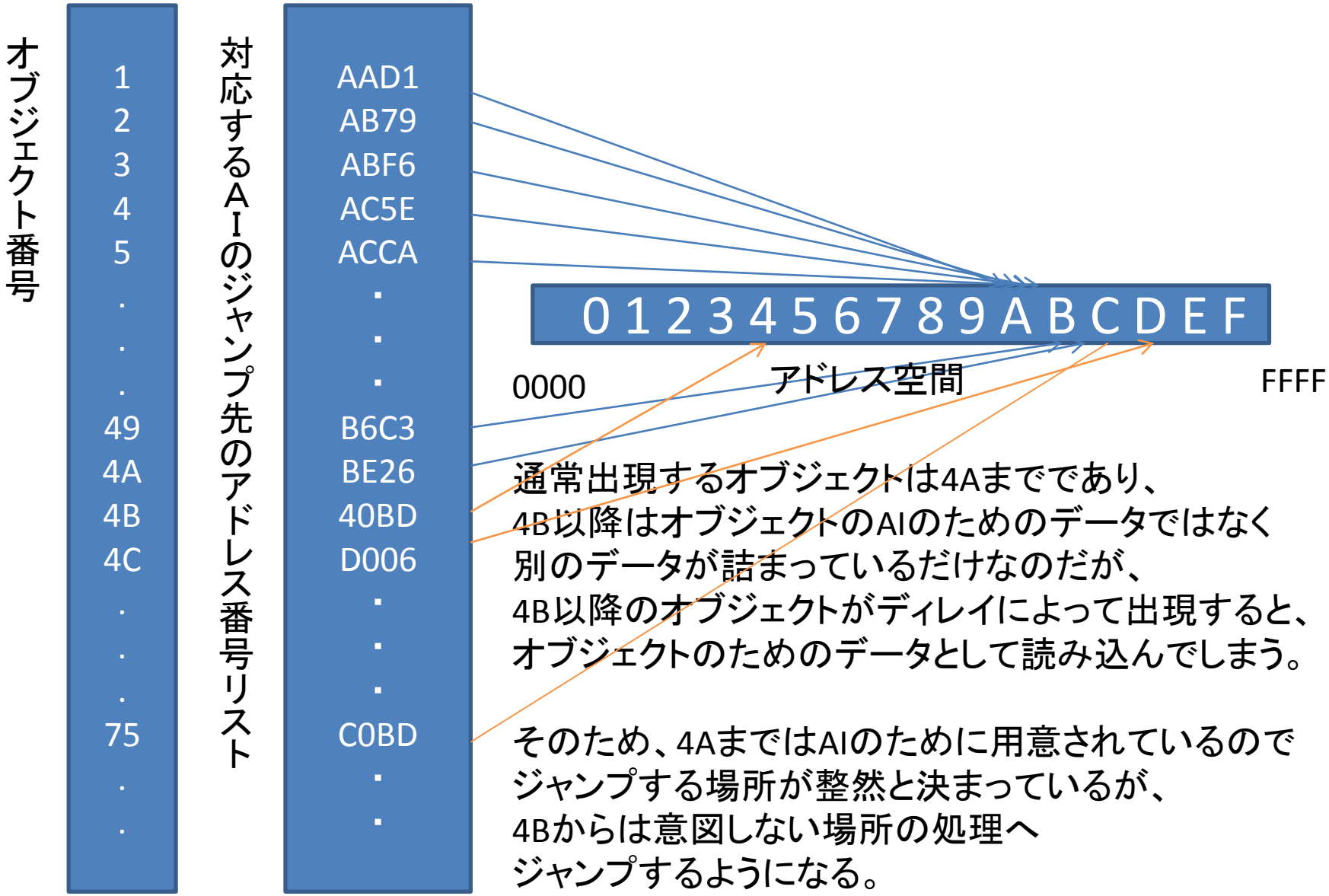
バンク5のデータを読み込んだ結果狂うのは、
「敵の情報を読み込むためのデータ」の場所を表すための情報

あるステージのあるマップにいるとき、
バンク5でデータが狂ったときの読み込むメモリは以下のようなイメージになる。
読み込もうとするメモリの位置はマップを移動しないと変化しない。



つまり、該当の範囲にあるメモリのデータがよく変動する場合、
色んなオブジェクトが出たりするし、
変動しにくいメモリの場合、固定のオブジェクトしか出なかったりする。

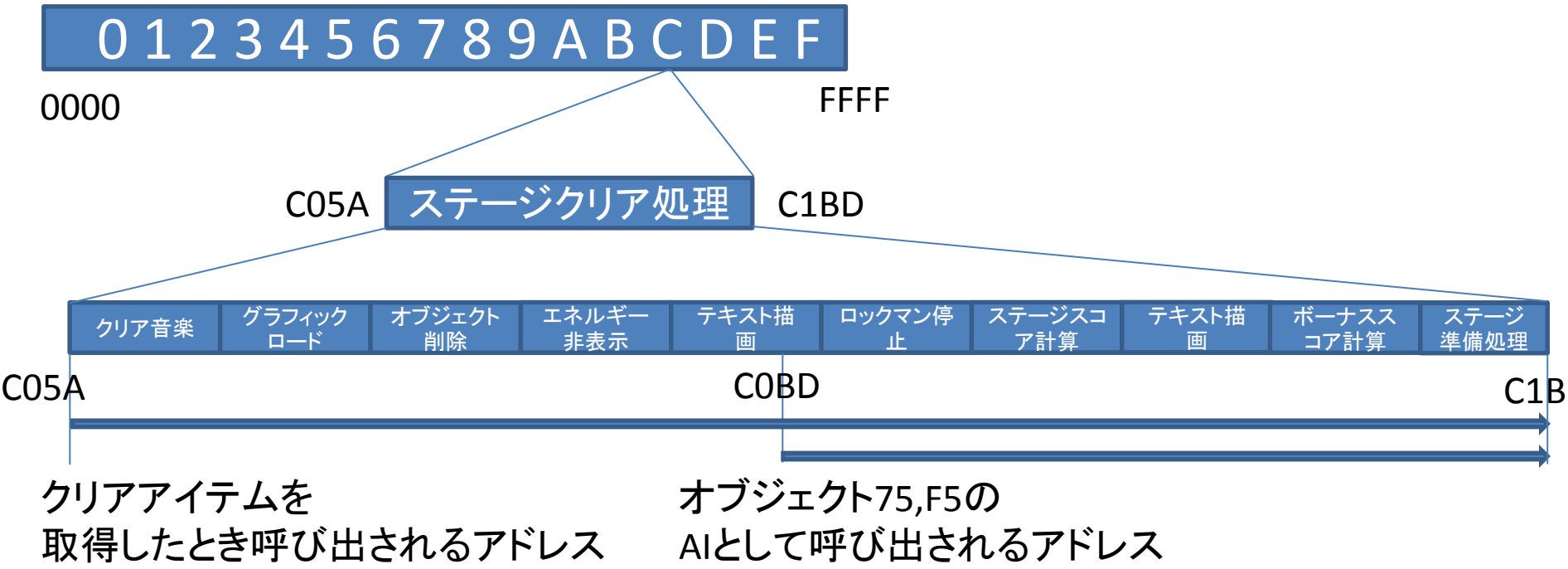
なぜオブジェクトのAIによって問題が起きるのか？



通常出現するオブジェクトは4Aまでであり、
4B以降はオブジェクトのAIのためのデータではなく
別のデータが詰まっているだけなのだが、
4B以降のオブジェクトがディレイによって出現すると、
オブジェクトのためのデータとして読み込んでしまう。

そのため、4AまではAIのために用意されているので
ジャンプする場所が整然と決まっているが、
4Bからは意図しない場所の処理へ
ジャンプするようになる。

ディレイステージクリア（オブジェクト75,F5)はどうやって起きるのか



オブジェクト75やF5のAIのジャンプ先として読み込まれるC0BDは、
たまたま運よくステージクリアの途中のアドレスである。

つまりディレイステージクリアは、ステージクリアの途中の処理が
オブジェクト75やF5のAIによっていきなり呼び出されることで発生する。

ディレイステージクリア（オブジェクト75,F5）まとめ

①非常に良い位置でNMI発生により、
バンク5から敵読み込み

1秒＝60フレーム

1フレーム＝10000命令

1フレーム

NMI

アイテム

ロックマン

敵

NMI

敵

$42 =5

A=6

バンクAへ切り替え

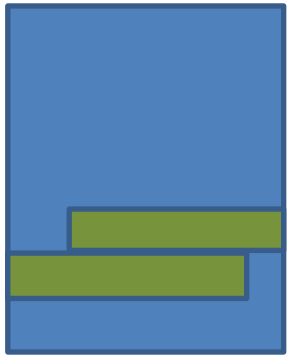
NMI

バンク$42へ切り替え

$42 =6

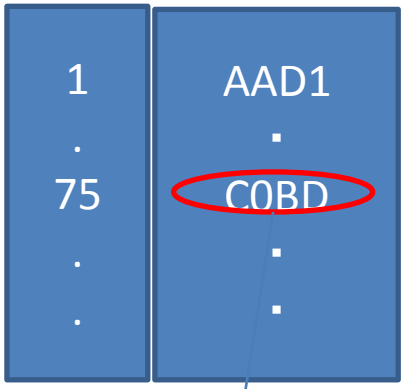
バンク5のまま
敵読み込み処理へ

②ステージのマップによって
決まる読み込み先の
データから
オブジェクト75かF5が発生



③75かF5の
AIが実行され、
ジャンプ先が決まる

オブジェクト番号

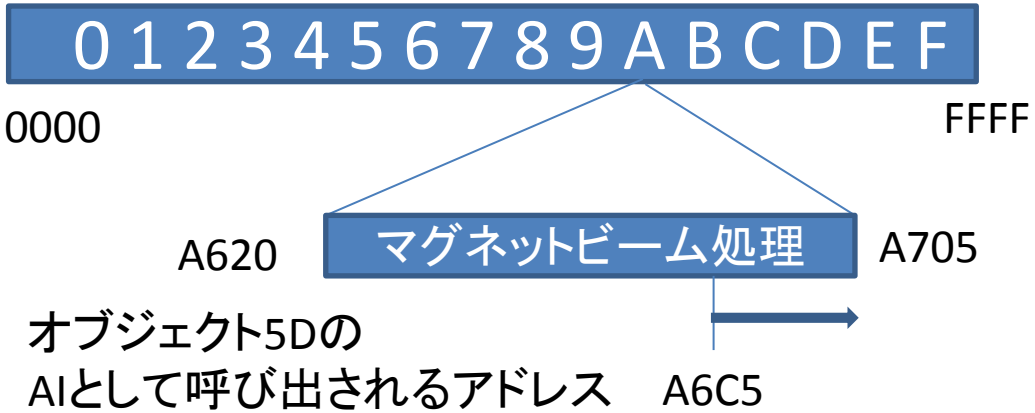


C05A ステージクリア処理 C1BD

C0BD

④ステージクリア処理の
途中から処理が実行され
ステージクリアとなる。

ディレイステージクリア（オブジェクト5D,DD)はどうやって起きるのか



オブジェクト5DやDDのAIのジャンプ先として読み込まれるA6C5は、
たまたま運よくマグネットビームの処理の途中のアドレスで、
この処理が途中から実行されると、$610～$6FFあたりのメモリが狂います。

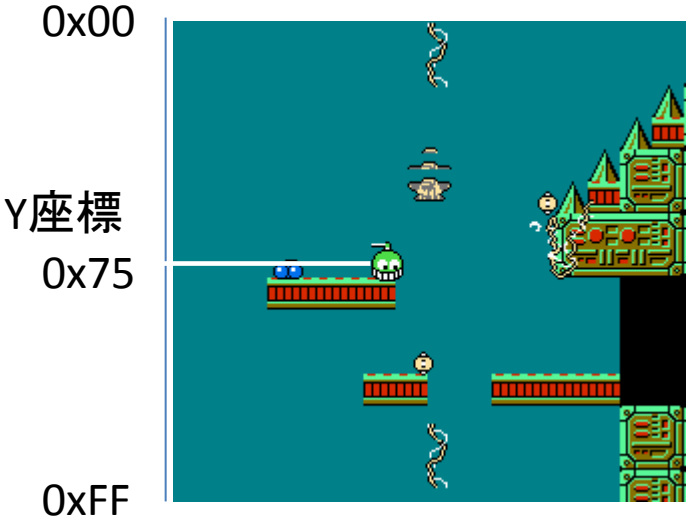
```
000600: 51 F8 F8 F8 F8 60 64 70 7C 70 10 D8 F8 F8 F8 F8
000610: 00 00 00 00 BA 00 7C 7C 75 AA 5A 7C 7A 7A 7A 0C
000620: 4A 80 00 00 C0 00 00 00 00 00 00 C0 00 A0 00 00
000630: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000640: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000650: 08 08 08 05 08 08 38 38 1A 64 5D 38 37 37 35 C9
000660: 40 00 00 00 40 00 00 00 00 00 00 00 A0 A0 60 00
000670: FD FD FD FE FD FD 40 40 00 40 40 40 40 40 40 04
000680: FE 05 00 00 FF 00 00 00 00 00 06 FA FE FE 01 00
000690: 2C 2C 2C 00 2C 2C FF FF FF FF FF FF FF FF FF 00
0006A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0006B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0006C0: 00 00 14 14 14 14 00 00 00 00 00 00 00 14 00 00
0006D0: 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14
0006E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F8 00
0006F0: 00 00 00 0F 00 00 3C 3C 00 3E 3E 3C 3D 3D 3E 5D
```

5DのAI実行前

```
000600: 00 F7 F7 F7 F7 60 60 70 7C 70 0F D7 F7 F7 F7 F7
000610: 00 00 00 00 F7 00 7B 7B 75 AA 59 7B 79 79 79 F7
000620: 89 7F 00 00 BF 00 7C 7C 75 AA 00 00 BF 00 9F 00
000630: 00 00 00 00 00 00 00 00 75 AA 00 00 00 00 00 00
000640: 00 00 00 00 00 00 00 00 75 A9 00 00 00 00 00 F8
000650: 00 00 00 00 00 00 00 00 75 00 00 00 00 00 00 F7
000660: 00 00 00 00 00 00 00 00 75 00 00 00 00 00 00 F7
000670: 00 00 FC FD 00 FC 00 00 75 00 3F 3F 00 3F 00 04
000680: FC 04 00 00 FE F7 F7 F7 75 A9 05 F9 FD FD 00 00
000690: 2B 2B 2B 00 F7 2B F7 F7 75 A9 F7 F7 F7 F7 F7 00
0006A0: 00 00 00 00 F6 00 F6 F6 75 00 00 00 F6 00 F6 00
0006B0: 01 01 00 00 01 01 01 01 75 01 01 01 01 01 00 00
0006C0: 01 01 02 01 15 02 01 01 76 01 00 02 01 02 00 00
0006D0: 00 00 14 14 15 02 01 01 76 00 02 02 01 01 F9 F8
0006E0: 00 00 14 00 14 01 00 00 75 00 01 01 00 01 F8 F8
0006F0: 00 00 00 0F 14 01 00 00 75 F8 01 01 00 3D F8 F8
```

5DのAI実行後

範囲内のメモリが大きく狂っていることがわかる



図：座標0x75の敵

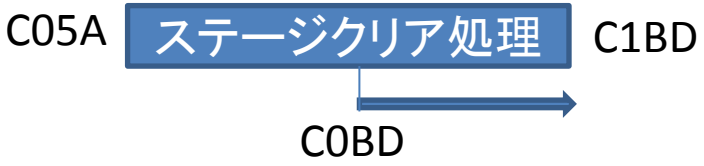
```
000600: 00 F7 F7 F7 F7 $618 0 7C 70 0F D7 F7 F7 F7 F7
000610: 00 00 00 00 F7       B 75 AA 59 7B 79 79 79 F7
000620: 89 7F 00 00 BF 00 7C 7C 75 AA 00 00 BF 00 9F 00
000630: 00 00 00 00 00 00 00 00 75 AA 00 00 00 00 00 00
000640: 00 00 00 00 00 00 00 00 75 A9 00 00 00 00 00 F8
000650: 00 00 00 00 00 00 00 00 75 00 00 00 00 00 00 F7
000660: 00 00 00 00 00 00 00 00 75 00 00 00 00 00 00 F7
000670: 00 00 FC FD 00 FC 00 00 75 00 3F 3F 00 3F 00 04
000680: FC 04 00 00 FE       7 75 A9 05 F9 FD FD 00 00
000690: 2B 2B 2B 00 F7 $6A8  7 75 A9 F7 F7 F7 F7 F7 00
0006A0: 00 00 00 00 F6       6 75 00 00 00 F6 00 F6 00
0006B0: 01 01 00 00 01 01 01 01 75 01 01 01 01 01 00 00
0006C0: 01 01 02 01 15 02 01 01 76 01 00 02 01 02 00 00
0006D0: 00 00 14 14 15 02 01 01 76 00 02 02 01 01 F9 F8
0006E0: 00 00 14 00 14 01 00 00 75 00 01 01 00 01 F8 F8
0006F0: 00 00 00 0F 14 01 00 00 75 F8 01 01 00 3D F8 F8
```

5DのAI実行後

$618の0x75の値が$6A8まで
コピーされているのがわかる

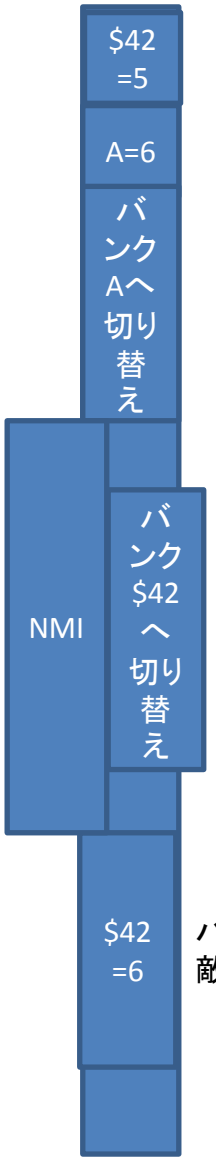
$618は、9番目の敵のY座標のメモリで、この値が75になっているとき、
オブジェクト5DのAIが実行されると、$6A8が75になることがある。

$6A0～6AFは敵のAIを実行するときのオブジェクト番号を保存するメモリであり
$6A8に75がコピーされると、75のAIが実行される。
すなわち、75のAIのジャンプ先であるC0BDが呼び出され、ステージクリアとなる。

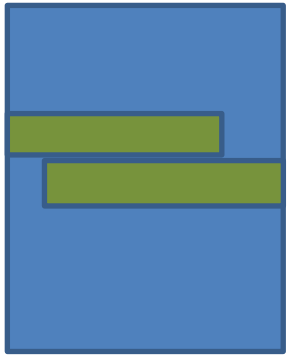


ディレイステージクリア（オブジェクト5D,DD）まとめ

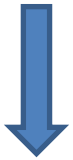
①非常に良い位置でNMI発生により、
バンク5から敵読み込み



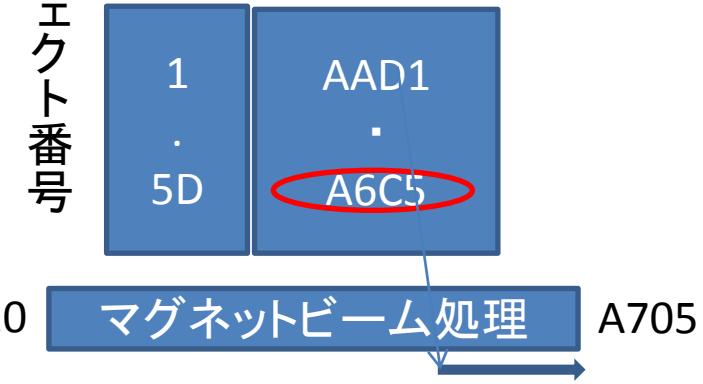
②ステージのマップによって
決まる読み込み先の
データから
オブジェクト5DかDDが発生



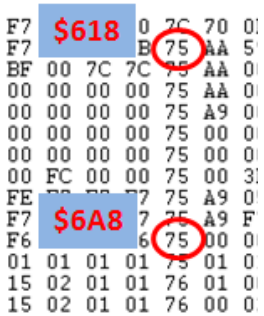
バンク5のまま
敵読み込み処理へ



③5DかDDの
AIが実行され、
ジャンプ先が決まる



④マグネットビームの処理が
途中から実行されることで、
敵のY座標=75が
AIのオブジェクト番号を保存する
メモリ$6A8へコピーされる



⑤75のAIが実行され、
ジャンプ先が決まる

⑥ステージクリアとなる。 C0BD