|  |  |
| --- | --- |
| struct M2Bone  {  int32 keyboneid;  int32 flags;  int16 parent;  int16 geoid;  int32 unknown;   |  | | --- | | AnimationBlock translation; |   AnimationBlock rotation;  AnimationBlock scaling;  Vec3D pivot;  }; |

|  |
| --- |
| 一个动画，对应一个\*.anim文件  struct ModelAnimationWotLK  {  int16 animID;  int16 subAnimID;  uint32 length;  float moveSpeed;  uint32 flags;  uint16 probability; uint16 unused;  uint32 d1;  uint32 d2;  uint32 playSpeed;  Sphere boundSphere;  int16 NextAnimation;  int16 Index;  }; |

|  |
| --- |
| struct AnimationBlockHeader  {  uint32 nEntrys;  uint32 ofsEntrys;  };  ofsEntrys可能在.anim文件中，也可能在.skin文件中  具体在哪根据ofsEntrys和anim文件size比较得到（这里不知道为啥做的如此诡异） |

|  |  |
| --- | --- |
| struct AnimationBlock  {  int16 type;  int16 seq;  uint32 nTimes;  uint32 ofsTimes;   |  | | --- | | uint32 nKeys;  uint32 ofsKeys; |   }; |

|  |
| --- |
| nKeys是AnimationBlockHeader的数量，  每个AnimationBlockHeader代表一个动作的一组关键帧 |

|  |
| --- |
| 一一对应 |

for(size\_t j=0; j < b.nTimes; j++) {

AnimationBlockHeader\* pHeadTimes = (AnimationBlockHeader\*)(f.getBuffer() + b.ofsTimes + j\*sizeof(AnimationBlockHeader));

uint32 \*ptimes;

if (animfiles[j].getSize() > pHeadTimes->ofsEntrys)

ptimes = (uint32\*)(animfiles[j].getBuffer() + pHeadTimes->ofsEntrys);

else if (f.getSize() > pHeadTimes->ofsEntrys)

ptimes = (uint32\*)(f.getBuffer() + pHeadTimes->ofsEntrys);

else

continue;

for (size\_t i=0; i < pHeadTimes->nEntrys; i++)

times[j].push\_back(ptimes[i]);

}