```
UInt32 PhysicsSystemG::RayCast(const Float3& origin, const Float3& unitDir, float maxDistance, CollisionChannelBit collisionChannel,
                HitFlag flag /*= HitFlag::Default*/, UInt32 maxHit/* = 1*/, PhysicsRayCastResult* outResults /*= nullptr*/)
{
 ......
    physx::PxQueryFilterData filterData;
    filterData.data.word1 = EncodeCollisionWord(false, CollisionTypeValue{}, collisionChannel);
    physx::PxRaycastBuffer hit(touchBuffer.get(), (maxHit == 1 ? 0 : maxHit));
    bool result = mPxScene->raycast(CrossVec3ToPxVec3(origin), CrossVec3ToPxVec3(unitDir), maxDistance, hit, pxFlag,
      filterData, nullptr, /*(maxHit == 1 && mRayCastCache.shape ? &mRayCastCache : nullptr)*/ nullptr);

    if (outResults)
    {
      if (maxHit == 1)
      {
          outResults->flags = PxHitFlagToCrossHitFlag(hit.block.flags);
          outResults->position = PxVec3ToCrossVec3(hit.block.position);
          outResults->uv = Float2A{ hit.block.u, hit.block.v };
          outResults->normal = PxVec3ToCrossVec3(hit.block.normal);
          outResults->distance = hit.block.distance;
          ......
       }
     }
}
```