

# Forum:[Hack Forums](#)

Thread: [C++ rust cheat code source](#)

**critical9** - February 28, 2024, 11:08 am

(not mine)

Rust Alpha win.799.81 - Official Rust - Devblog 98

Game Object Manager:

Code #define GAME\_OBJECT\_MANAGER 0x125D5C0

```
class GameObjectManager
{
public:
    LastObjectBase* lastTaggedObject; //0x0000
    BaseObject* taggedObjects; //0x0008
    LastObjectBase* lastActiveObject; //0x0010
    BaseObject* activeObjects; //0x0018
};
```

```
class BaseObject
{
public:
    char pad_0x0000[0x8]; //0x0000
    BaseObject* nextObjectLink; //0x0008
    GameObject* object; //0x0010
};
```

```
class LastObjectBase
{
public:
    char pad_0x0000[0x10]; //0x0000
    GameObject* lastObject; //0x0010
};
```

Game Objects:

Code class GameObject

```
{
public:
    char pad_0x0000[0x8]; //0x0000
    __int32 m_instanceID; //0x0008
    char pad_0x000C[0x24]; //0x000C
    __int32 m_label; //0x0030
    char pad_0x0034[0x4]; //0x0034
    __int32 m_size; //0x0038
    char pad_0x003C[0x4]; //0x003C
    __int32 m_capacity; //0x0040
    char pad_0x0044[0x4]; //0x0044
    __int32 m_layer; //0x0048
    __int16 m_tag; //0x004C
    unsigned char m_isActive; //0x004E
};
```

```

    unsigned char m_isActiveCached; //0x004F
    unsigned char m_isDestroying; //0x0050
    unsigned char m_isActivating; //0x0051
    char pad_0x0052[0x6]; //0x0052
    String* m_objectName; //0x0058
};

```

Each GameObject also has a pointer to it's corresponding classes at 0x28.

Transform:

Code GameObject[ + 0x28] + 0x8 = Transform

The following Transform works for most GameObject's except for MISC( objectTag 20006 ).

Code class Transform

```

{
public:
    char pad_0x0000[0x34]; //0x0000
    Vector3 rotation; //0x0034
    Vector3 origin; //0x0040
};

```

class MiscTransform

```

{
public:
    char pad_0x0000[0x8C]; //0x0000
    D3DXVECTOR3 origin; //0x008C
};

```

MainCamera (objectTag 5):

Code GameObject[ + 0x28] = CameraObject

class CameraObject

```

{
public:
    char pad_0x0000[0x8]; //0x0000
    Transform* transform; //0x0008
    char pad_0x0010[0x8]; //0x0010
    Camera* camera; //0x0018
};

```

class Camera

```

{
public:
    char pad_0x0000[0xB8]; //0x0000
    D3DXMATRIX viewMatrix; //0x00B8
};

```

PlayerObject (objectTag 6):

Code class PlayerObject

```

{
public:
    char pad_0x0000[0x8]; //0x0000
    Transform* transform; //0x0008
    char pad_0x0010[0x8]; //0x0010
    Entity* entity; //0x0018
};

```

PlayerObject] + 0x18] + 0x20] = BaseEntity

```
class BaseEntity
{
public:
    char pad_0x0000[0x90]; //0x0000
    DWORD64 moreShit; //0x0090
    char pad_0x0098[0xB0]; //0x0098
    unsigned char dead; //0x0148
    char pad_0x0149[0x3]; //0x0149
    float m_health; //0x014C
    float m_previousHealth; //0x0150
    char pad_0x0154[0x14]; //0x0154
    DWORD64 moreee; //0x0168
    char pad_0x0170[0x18]; //0x0170
    LookingAtPtr* m_lookingAt; //0x0188
    BaseEntity* m_lookingAtEntity; //0x0190
    DWORD64 m_lookingAtCollider; //0x0198
    char pad_0x01A0[0x20]; //0x01A0
    PlayerInventory* m_playerInventory; //0x01C0
    char pad_0x01C8[0x18]; //0x01C8
    PlayerWalkMovement* m_playerWalkMovement; //0x01E0
    char pad_0x01E8[0x18]; //0x01E8
    UnicodeString* m_steamIDString; //0x0200
    UnicodeString* m_playerNameString; //0x0208
    char pad_0x0210[0x20]; //0x0210
    unsigned char cameraTypeBackup; //0x0230
    char pad_0x0231[0x3]; //0x0231
    unsigned char cameraTypeReal; //0x0234
    char pad_0x0235[0x3]; //0x0235
    unsigned char inFreeCamMode; //0x0238
    char pad_0x0239[0x3]; //0x0239
    D3DXVECTOR3 eyepos; //0x023C
    char pad_0x0248[0x4]; //0x0248
    D3DXVECTOR3 N000046E6; //0x024C
    char pad_0x0258[0xC]; //0x0258
    unsigned char playerFlags; //0x0264
    unsigned char isConnected; //0x0265
    char pad_0x0266[0x2]; //0x0266
    __int32 m_activeItemUID; //0x0268
    char pad_0x026C[0xC]; //0x026C
    __int64 m_steamID; //0x0278
};

class LookingAtPtr
{
public:
    char pad_0x0000[0x18]; //0x0000
    GameObject* m_object; //0x0018
};

class PlayerWalkMovement
{
public:
```

```

char pad_0x0000[0x28]; //0x0000
unsigned char adminCheat; //0x0028
char pad_0x0029[0x3F]; //0x0029
float N00004A49; //0x0068
float N00004568; //0x006C
float playerHeight; //0x0070
float playerCenter; //0x0074
float playerHeightDucked; //0x0078
float playerCenterDucked; //0x007C
float gravityTestRadius; //0x0080
float gravityMultiplier; //0x0084
float gravityMultiplierSwimming; //0x0088
float maxAngleWalking; //0x008C
float maxAngleClimbing; //0x0090
float maxAngleSliding; //0x0094
float maxVelocity; //0x0098
};

class UnicodeString
{
public:
    char pad_0x0000[0x10]; //0x0000
    __int32 size; //0x0010
    wchar_t str; //0x0014
};

Object Tags:
Code enum objects
{
    MAINCAMERA = 5,
    PLAYER = 6,
    TERRAIN = 20001,
    CORPSE = 20009, //Animals & Players
    MISC = 20006, //Trash cans, Sleeping Bags, Storage Box, etc
    ANIMAL = 20008,
    SKYDOME = 20012,
    RIVERMESH = 20014,
    MONUMENT = 20015 //Airport, Powerplant, etc
};

Filtering Objects:
Use m_layer from GameObject:
Code enum layers
{
    kDefaultLayer = 0,
    kNoFXLayer = 1,
    kIgnoreRaycastLayer = 2,
    kIgnoreCollisionLayer = 3,
    kWaterLayer = 4,
    kUILayer = 5,
    kDeployed = 8, //repair bench/research table/tool cupboard/etc
    kRagdoll = 9, //supply signal/heli crates
    kInvisible = 10,
    kAI = 11,
    kPlayerMovement = 12,

```

```

kSkyReflection = 13,
kSky = 15,
kWorld = 16,
kServerPlayer = 17,
kTrigger = 18,
kViewModel = 20,
kConstruction = 21,
kTerrain = 23,
kDebris = 26, //dropped items
kDisplacement = 27,
kTree = 30
};

```

WorldToScreen:

Code bool WorldToScreen(D3DXVECTOR3 origin, D3DXVECTOR2 \* out)

```

{
    D3DXMATRIX temp;

    D3DXMatrixTranspose(&temp, &viewMatrix);

    D3DXVECTOR3 translationVector = D3DXVECTOR3(temp._41, temp._42, temp._43);
    D3DXVECTOR3 up = D3DXVECTOR3(temp._21, temp._22, temp._23);
    D3DXVECTOR3 right = D3DXVECTOR3(temp._11, temp._12, temp._13);

    float w = D3DXVec3Dot(&translationVector, &origin) + temp._44;

    if (w < 0.098f)
        return false;

    float y = D3DXVec3Dot(&up, &origin) + temp._24;
    float x = D3DXVec3Dot(&right, &origin) + temp._14;

    out->x = (res[0] / 2) * (1.f + x / w);
    out->y = (res[1] / 2) * (1.f - y / w);

    return true;
}

```

Player Inventory:

Code PlayerInfo + 0x1B8] = PlayerInventory

PlayerInfo + 0x260] = m\_activeItemUID

Code:

```

class PlayerInventory
{
public:
    char pad_0x0000[0x28]; //0x0000
    ItemContainer* m_containerMain; //0x0028
    ItemContainer* m_belt; //0x0030
    ItemContainer* m_wear; //0x0038
};

```

class ItemContainer

```

{
public:
    char pad_0x0000[0x20]; //0x0000
    ItemList* m_itemList; //0x0020
    char pad_0x0028[0x8]; //0x0028
    PlayerInfo* entityOwner; //0x0030
    char pad_0x0038[0x10]; //0x0038
    __int32 capacity; //0x0048
};

class ItemList
{
public:
    char pad_0x0000[0x10]; //0x0000
    Items* m_items; //0x0010
    __int32 m_size; //0x0018
};

class Items
{
public:
    char pad_0x0000[0x20]; //0x0000
    Item* m_items[24]; //0x0020
};

class ItemDefinition
{
public:
    char pad_0x0000[0x20]; //0x0000
    UnicodeString* m_displayName; //0x0020
    char pad_0x0028[0x60]; //0x0028
    __int32 m_ItemID; //0x0088
    char pad_0x008C[0x1C]; //0x008C
    __int32 m_rarity; //0x00A8
};

class Item
{
public:
    char pad_0x0000[0x18]; //0x0000
    ItemDefinition* m_ItemDefinition; //0x0018
    char pad_0x0020[0x40]; //0x0020
    float m_condition; //0x0060
    float m_maxCondition; //0x0064
    __int32 m_uid; //0x0068
    char pad_0x006C[0x4]; //0x006C
    __int32 m_amount; //0x0070
    __int32 m_slot; //0x0074
};
};
Aimbot:
Code Inside BasePlayer/BaseEntity:

```

```
PlayerInput* input; //0x01E0
```

```
class PlayerInput
```

```
{
```

```
public:
```

```
    char pad_0x0000[0x44]; //0x0000
```

```
    Vector4 bodyRotation; //0x0044
```

```
    Vector3 bodyAngles; //0x0054 //write to this
```

```
    Vector4 headRotation; //0x0060
```

```
    Vector3 headAngles; //0x0070
```

```
    Vector2 viewDelta; //0x007C
```

```
    Vector3 headDelta; //0x0084
```

```
    __int32 mouseWheelUp; //0x0090
```

```
    __int32 mouseWheelDn; //0x0094
```

```
    char pad_0x0098[0x13C]; //0x0098
```

```
}; //Size=0x01D4
```

```
coded 12 december 2015
```