

1.1 キーボードのAが押されたらプレイヤーの移動速度を倍にしてみよう。

まず、キーボードのAが押されたということを記録できるようにする必要があります。

Packman のプロジェクトでキーボードの入力を記録しているプログラムは下記の二のファイルです。

ソースファイル

.¥Packman¥tkEngine¥Input¥tkInput¥tkKeyInput.cpp

ヘッダーファイル

.¥Packman¥tkEngine¥Input¥tkInput¥tkKeyInput.h

ではヘッダーファイルを下記のように編集してください。太字になっている部分が変更点です。

```

/*!
 * @brief      キー入力。
 */

#ifndef _TKKEYINPUT_H_
#define _TKKEYINPUT_H_

namespace tkEngine{
    class CKeyInput{
        enum EnKey{
            enKeyUp,
            enKeyDown,
            enKeyRight,
            enKeyLeft,
            enKeyA,
            enKeyNum,
        };
    public:
        /*!
         * @brief      コンストラクタ。
         */
        CKeyInput();
        /*!
         * @brief      デストラクタ。
         */
        ~CKeyInput();
        /*!
         * @brief      キー情報の更新。
         */
        void Update();
        /*!
         * @brief      上キーが押されている。
         */
        bool IsUpPress() const
        {
            return m_keyPressFlag[enKeyUp];
        }
        /*!
         * @brief      右キーが押されている。
         */
        bool IsRightPress() const
        {
            return m_keyPressFlag[enKeyRight];
        }
    };
}

```

```

        * @brief      左キーが押されている。
        */
    bool IsLeftPress() const
    {
        return m_keyPressFlag[enKeyLeft];
    }
    /*!
    * @brief      下キーが押されている。
    */
    bool IsDownPress() const
    {
        return m_keyPressFlag[enKeyDown];
    }
    /*!
    * @brief      キーボードの A が押された。
    */
    bool IsAPress() const
    {
        return m_keyPressFlag[enKeyA];
    }
private:
    bool m_keyPressFlag[enKeyNum];
};
}
#endif // _TKKEYINPUT_H_

```

続いてソースファイルを下記のように変更します。

```

/*!
 * @brief      キー入力
 */

#include "tkEngine/tkEnginePreCompile.h"
#include "tkEngine/Input/tkKeyInput.h"

namespace tkEngine{
    /*!
    * @brief      コンストラクタ。
    */
    CKeyInput::CKeyInput()
    {
        memset(m_keyPressFlag, 0, sizeof(m_keyPressFlag));
    }
    /*!
    * @brief      デストラクタ。
    */
    CKeyInput::~~CKeyInput()
    {
    }
    /*!
    * @brief      キー情報の更新。
    */
    void CKeyInput::Update()
    {
        if (GetAsyncKeyState(VK_UP) & 0x8000) {
            m_keyPressFlag[enKeyUp] = true;
        }
        else {
            m_keyPressFlag[enKeyUp] = false;
        }
        if (GetAsyncKeyState(VK_DOWN) & 0x8000) {
            m_keyPressFlag[enKeyDown] = true;
        }
        else {

```

```

        m_keyPressFlag[enKeyDown] = false;
    }
    if (GetAsyncKeyState(VK_RIGHT) & 0x8000) {
        m_keyPressFlag[enKeyRight] = true;
    }
    else {
        m_keyPressFlag[enKeyRight] = false;
    }
    if (GetAsyncKeyState(VK_LEFT) & 0x8000) {
        m_keyPressFlag[enKeyLeft] = true;
    }
    else {
        m_keyPressFlag[enKeyLeft] = false;
    }
    if ((GetAsyncKeyState('A') & 0x8000)
        | (GetAsyncKeyState('a') & 0x8000)) {
        m_keyPressFlag[enKeyA] = true;
    }
    else {
        m_keyPressFlag[enKeyA] = false;
    }
}
}

```

これでキーボードの A が入力されると、m_keyPressFlag[enKeyA]に true という値が記録されるようになりました。

では本当にキーボードの A が入力されたら true が設定されるか確認してみましょう。先ほどの書き換えた部分にコードを下記の黒字のコードを追加してみてください。

```

if ((GetAsyncKeyState('A') & 0x8000)
    | (GetAsyncKeyState('a') & 0x8000)) {
    m_keyPressFlag[enKeyA] = true;
    MessageBox(NULL, "A ボタンが押されたよ！", "成功", MB_OK);
}
else {
    m_keyPressFlag[enKeyA] = false;
}

```

A を押したらダイアログボックスがでましたか？

では、確認ができたならメッセージボックスを表示するコードは削除してください。

では、正しくキーボードから入力を受け取ることができることが確認できたのでプレイヤーの移動速度を倍にしてみましょう。プレイヤーの移動処理は下記のファイルに記述されています。

ソースファイル

Packman¥Packman¥game¥Player¥ CPlayer.cpp

ヘッダーファイル

Packman¥Packman¥game¥Player¥ CPlayer.h

CPlayer.cpp を下記のように書き換えてみて下さい。

```

/*!
 * @brief      プレイヤー
 */

#include "stdafx.h"
#include "Packman/game/Player/CPlayer.h"
#include "Packman/game/CGameManager.h"

/*!
 * @brief      Update が初めて呼ばれる直前に一度だけ呼ばれる処理。
 */
void CPlayer::Start()
{
}

/*!
 * @brief      Update 関数が実行される前に呼ばれる更新関数。
 */
void CPlayer::PreUpdate()
{
    Move();
}

/*!
 * @brief      更新処理。60fps なら 16 ミリ秒に一度。30fps なら 32 ミリ秒に一度呼ばれる。
 */
void CPlayer::Update()
{
    m_sphere.SetPosition(m_position);
    m_sphere.UpdateWorldMatrix();
    CGameManager& gm = CGameManager::GetInstance();
    CMatrix mMVP = gm.GetGameCamera().GetViewProjectionMatrix();
    const CMatrix& mWorld = m_sphere.GetWorldMatrix();
    m_wvpMatrix.Mul(mWorld, mMVP);
    m_idMapModel.SetWVPMatrix(m_wvpMatrix);
    IDMap().Entry(&m_idMapModel);
    m_shadowModel.SetWorldMatrix(mWorld);
    ShadowMap().Entry(&m_shadowModel);
}

/*!
 * @brief      移動処理。
 */
void CPlayer::Move()
{
    float moveSpeed = 0.02f; //移動速度。
    if (KeyInput().IsAPress()) {
        //キーボードの A 押されていたら速度を倍にする。
        moveSpeed *= 2.0f;
    }
    if (KeyInput().IsUpPress()) {
        m_position.z += moveSpeed;
    }
    if (KeyInput().IsDownPress()) {
        m_position.z -= moveSpeed;
    }
    if (KeyInput().IsRightPress()) {
        m_position.x += moveSpeed;
    }
    if (KeyInput().IsLeftPress()) {
        m_position.x -= moveSpeed;
    }
}

/*!
 * @brief      描画処理。60fps なら 16 ミリ秒に一度。30fps なら 32 ミリ秒に一度呼ばれる。
 */
void CPlayer::Render(tkEngine::CRenderContext& renderContext)
{
    CGameManager& gm = CGameManager::GetInstance();
    m_sphere.RenderLightWVP(
        renderContext,

```

```
        m_wvpMatrix,
        gm.GetFoodLight(),
        false,
        true
    );
}
/*!
 * @brief 構築。
 * 必ず先に CreateShape を一度コールしておく必要がある。
 */
void CPlayer::Build( const CVector3& pos )
{
    m_sphere.Create(0.08f, 10, 0xffff0000, true );
    m_idMapModel.Create(m_sphere.GetPrimitive());
    m_shadowModel.Create(m_sphere.GetPrimitive());
    m_position = pos;
}
```

チャレンジ

では先ほどのプログラムを改造して A ボタンが押されたらパックマンがジャンプする
ようにしてみましょう。