```
#include "Player.h"
#include "../Utility/InputControl.h"
#include "Dxlib.h"

Player::Player() : is_active(false), image(NULL), location(0.0f), box_size(0.0f),
angle(0.0f),
                           speed(0.0f), hp(0.0f), fuel(0.0f), barrier_count(0),
barrier(nullptr)
{

}

Player::~Player()
{

}

// 初期化処理
void Player::Initialize()
{
      is_active = true;
      location = Vector2D(320.0f, 380.0f);
      box_size = Vector2D(31.0f, 60.0f);
      angle = 0.0f;
      speed = 3.0f;
      hp = 1000;
      fuel = 20000;
      barrier_count = 3;

      // 画像の読み込み
      image = LoadGraph("Resource/images/car1pol.bmp");

      // エラーチェック
      if (image == -1)
      {
            throw ("Resource/images/car1pol.bmpがありません\n");
      }
}
```

```
// 更新処理
void Player::Update()
{
	// 操作不可状態であれば、自身を回転させる
	if (!is_active)
	{
		angle += DX_PI_F / 24.0f;
		speed = 1.0f;
		if (angle >= DX_PI_F * 4.0f)
		{
			is_active = true;
		}
		return;
	}

	// 燃料の消費
	fuel -= speed;

	// 移動処理
	Movement();

	// 加減速処理
	Acceleration();

	if (InputControl::GetButtonDown(XINPUT_BUTTON_START))
	{
		is_active = false;
	}

	// バリア処理
	if (InputControl::GetButtonDown(XINPUT_BUTTON_B) && barrier_count > 0)
	{
		if (barrier == nullptr)
		{
			barrier_count--;
			barrier = new Barrier;
		}
	}
```

```
	// バリアが生成されていたら、更新を行う
	if (barrier != nullptr)
	{
		// バリア時間が経過したか？していたら、削除する
		if (barrier->IsFinished(this->speed))
		{
			delete barrier;
			barrier = nullptr;
		}
	}
}

// 描画処理
void Player::Draw()
{
	// プレイヤー画像の描画
	DrawRotaGraphF(location.x, location.y, 1.0, angle, image, TRUE);

	// バリアが生成されていたら、描画を行う
	if (barrier != nullptr)
	{
		barrier->Draw(this->location);
	}
}

// 終了時処理
void Player::Finalize()
{
	// 読み込んだ画像を削除
	DeleteGraph(image);

	// バリアが生成されていたら、削除する
	if (barrier != nullptr)
	{
		delete barrier;
	}
}

// 状態設定処理
```

```
void Player::SetActive(bool flg)
{
	this->is_active = flg;
}

// 体力減少処理
void Player::DecreaseHp(float value)
{
	this->hp += value;
}

// 位置情報取得処理
Vector2D Player::GetLocation() const
{
	return this->location;
}

// 当たり判定の大きさ取得処理
Vector2D Player::GetBoxSize() const
{
	return this->box_size;
}

// 速さ取得処理
float Player::GetSpeed() const
{
	return this->speed;
}

// 燃料取得処理
float Player::GetFuel() const
{
	return this->fuel;
}

// 体力取得処理
float Player::GetHp() const
{
	return this->hp;
```

```
}

// バリア枚数取得処理
int Player::GetBarriarCount() const
{
	return this->barrier_count;
}

// バリア有効か？を取得
bool Player::IsBarrier() const
{
	return (barrier != nullptr);
}

// 移動処理
void Player::Movement()
{
	Vector2D move = Vector2D(0.0f);
	angle = 0.0f;

	// 十字移動処理
	if (InputControl::GetButton(XINPUT_BUTTON_DPAD_LEFT))
	{
		move += Vector2D(-1.0f, 0.0f);
		angle = -DX_PI_F / 18;
	}
	if (InputControl::GetButton(XINPUT_BUTTON_DPAD_RIGHT))
	{
		move += Vector2D(1.0f, 0.0f);
		angle = DX_PI_F / 18;
	}
	if (InputControl::GetButton(XINPUT_BUTTON_DPAD_UP))
	{
		move += Vector2D(0.0f, -1.0f);
	}
	if (InputControl::GetButton(XINPUT_BUTTON_DPAD_DOWN))
	{
		move += Vector2D(0.0f, 1.0f);
	}
```

```
	location += move;

	// 画面外に行かないように制限する
	if ((location.x < box_size.x) || (location.x >= 640.0f - 180.0f) ||
		(location.y < box_size.y) || (location.y >= 480.0f - box_size.y))
	{
		location -= move;
	}
}

// 加減速処理
void Player::Acceleration()
{
	// LBボタンが押されたら、減速する
	if (InputControl::GetButtonDown(XINPUT_BUTTON_LEFT_SHOULDER) && speed >
1.0f)
	{
		speed -= 1.0f;
	}

	// RBボタンが押されたら、加速する
	if (InputControl::GetButtonDown(XINPUT_BUTTON_RIGHT_SHOULDER) && speed <
10.0f)
	{
		speed += 1.0f;
	}
}
```