内容

[ちゃぷたー１　たいとる画面を作ろう 2](#_Toc522459259)

[ちゃぷたー2　インゲームに遷移しよう 3](#_Toc522459260)

[ちゃぷたー3　たいとる画面に戻れるようにしよう 4](#_Toc522459261)

[ちゃぷたー４　インゲームでプレイヤーを表示しよう 6](#_Toc522459262)

[ちゃぷたー5　背景を表示しよう 8](#_Toc522459263)

[ちゃぷたー６　キャラを動かそう 9](#_Toc522459264)

[ちゃぷたー７　カメラをキャラに追従させよう 11](#_Toc522459265)

[ちゃぷたー８　キャラクターと背景のあたり判定を取ってみよう 13](#_Toc522459266)

[ちゃぷたー９　キャラクター重力を加えよう 15](#_Toc522459267)

[ちゃぷたー１０　Ａボタンでキャラクターをジャンプさせてみよう。 17](#_Toc522459268)

[ちゃぷたー１１　スターを表示しよう 19](#_Toc522459269)

[ちゃぷたー１２　スターを回してみよう 21](#_Toc522459270)

[ちゃぷたー１３　スターをゲットしてみよう 23](#_Toc522459271)

[ちゃぷたー１５　星を全部取れたらゲームクリア演出を行おう 25](#_Toc522459272)

はじめてのゲーム制作

# ちゃぷたー１　たいとる画面を作ろう

レシピ

1. IGameObjectを継承したTitleクラスを作成する。

2. TitleクラスでAssets/sprite/title.pngを表示するプログラムを書く。

3. Titleクラスのインスタンスをmain.cppで生成する。

どこでインスタンスを作るのか、しっかり考えるんじゃぞ



完成動画



参考資料

1. tkEngine2/Sample/Sample\_06

2. ゲームプログラミングの教材.pdfのLesson 17\_1～Lesson 17\_8

# ちゃぷたー2　インゲームに遷移しよう

レシピ

1. IGameObjectを継承したGameクラスを作成する。

2. TitleクラスにUpdate関数を追加して、Aボタンが押されたらGameクラスのインスタンスを作成する処理を実装する。

いらニャいインスタンスを捨てるのを忘れるニャよ。おれは捨てるニャよ！



完成動画



参考資料

１. ゲームプログラミングの教材.pdfのLesson 17\_1～Lesson 17\_8

# ちゃぷたー3　たいとる画面に戻れるようにしよう

レシピ

1. GameクラスにUpdate関数を追加。

2. GameクラスのUpdate関数の中で、コントローラーのバックボタンが押されたら、Titleクラスのインスタンスを作成する処理を実装して、タイトル画面に戻れるようにする。セレクトボタンが押されたかどうかは

下記のようなコードで判定できる

|  |
| --- |
| if( Pad(0).IsPress( enButtonSelect ) == true ) {  //ここにセレクトボタンが押されたときの処理を書くのじゃ。  } |

いらニャいインスタンスを捨てるのを忘れるニャよ。おれは捨てるニャよ！(大事なので２回言った)



完成動画



参考資料

１. ゲームプログラミングの教材.pdfのLesson 17\_1～Lesson 17\_8

# ちゃぷたー４　インゲームでプレイヤーを表示しよう

レシピ

1.　カメラが必要になるので、GameCameraクラスを作成する。

2.　GameCameraクラスのコンストラクタで、メインカメラに注視点、視点などを設定して更新(Update)する。

3.　Gameクラスのコンストラクタで、GameCamreaのインスタンスを作成する。

4.　Playerクラスを作成する。

5.　PlayerクラスにmodelData/unityChan.cmoファイルを表示する処理を追加する。

6.　Gameクラスのコンストラクタで、Playerクラスのインスタンスを作成する。

7. タイトル画面とインゲームを行き来して、リソースの開放忘れなどが起きていないか、しっかりと確認する。

NewGOで作ったインスタンスはDeleteGOで破棄できるニャ。



用語集

　コンストラクタ

　　→インスタンスが作られたときに自動的に呼ばれる関数。

デストラクタ

　　→インスタンスが破棄されるときに自動的に呼ばれる関数。

完成動画



参考資料

１.　 tkEngine2/Sample/Sample\_00

２.　ゲームプログラミングの教材.pdfのLesson\_00\_2～Lesson\_00\_3

３.　ゲームプログラミングの教材.pdfのLesson\_16\_7

# ちゃぷたー5　背景を表示しよう

レシピ

　１.　Backgroundクラスを作成して、modelData/background.cmoを表示できるようにする。

　２.　GameクラスのコンストラクタでBackgroundクラスのインスタンスを作成する。

　３.　例のごとくタイトル画面と行き来して、問題がないか確認する。



完成動画



参考資料

１.　 tkEngine2/Sample/Sample\_00

２.　ゲームプログラミングの教材.pdfのLesson\_00\_2～Lesson\_00\_3

３.　ゲームプログラミングの教材.pdfのLesson\_16\_7

# ちゃぷたー６　キャラを動かそう

レシビ

　１.　PlayerクラスにUpdate関数を追加

　２.　Playerクラスにプレイヤーの位置を表す、CVector3型のメンバ変数を追加。

　３.　PlayerクラスのUpdate関数で、パッドの入力に応じてキャラの座標を動かす処理を実装する。

座標をCSkinModelRenderに教えてやらないとキャラは動かニャいぞ！



完成動画



参考資料

　１．　tkEngine2/Sample/Sample\_01

# ちゃぷたー７　カメラをキャラに追従させよう

レシビ

　１.　GameCameraクラスからPlayerのインスタンスにアクセスできるようにする。

　２.　GameCameraクラスにUpdate関数を追加する。

　３.　PlayerクラスのUpdate関数で、パッドの入力に応じてキャラの座標を動かす処理を実装する。

インスタンスはFindGOを使えば見つけることができるニャ。花瓶を割った奴？しらニャいニャぁ(´・ω・｀)



完成動画



参考資料

１． tkEngine2/Sample/Sample\_08

２．　ゲームプログラミングの教材.pdfのLesson\_04～Lesson\_04\_05

３．　ゲームプログラミングの教材.pdfのLesson\_16\_5

# ちゃぷたー８　キャラクターと背景のあたり判定を取ってみよう

レシビ

　１．　BackgroudクラスにCPhysicsStaticObject型のメンバ変数を追加。

　２． BackgroundクラスのコンストラクタでCPhysicsStaticObjectの初期化を行う。

　　　　→CPhysicsStaticObjectは3Dモデルからあたり判定を行うための形状データ(Collider)

を作成してくれるぞ！

　 ３．　PlayerクラスにCCharacterController型のメンバ変数を追加。

　 ４．　Playerクラスのコンストラクタでキャラクターコントローラーの初期化を行う。

　 ５．　Player::Update関数に記述されている、キャラクターの移動処理を、速度ベースに変更

する。

レシピを見てもわけがわからん！！！という人は参考資料のSample/Sample\_07を読んでみるニャ！ちゃんと意味を考えて読むニャ。



完成動画



参考資料

　１．　tkEngine2/Sample/Sample\_07

　２．　ゲームプログラミングの教材.pdfのLesson\_05\_01

# ちゃぷたー９　キャラクター重力を加えよう

レシピ

　1. キャラクターの移動速度に重力加速度の影響を与えるようにする。

チャプター8のPlayer::Update関数にコードを1行追加するだけで重力落下は実装できるニャ。ゲームプログラミングの教材.pdfのLesson\_01\_04が参考になるけど、丸写ししても駄目ニャ！



完成動画



参考資料

　１．　ゲームプログラミングの教材.pdfのLesson\_01\_04

　２．　tkEngine2/Sample/Sample\_07

# ちゃぷたー１０　Ａボタンでキャラクターをジャンプさせてみよう。

レシピ

　1. Aボタンが押されたらキャラクターの移動速度のY方向に力を与える！

チャプター9のPlayer::Update関数にコードを３行追加するだけニャ！難しく考えニャい！



完成動画



参考資料

　１．　ゲームプログラミングの教材.pdfのLesson\_01\_03～Lesson\_01\_04

　２．　tkEngine2/Sample/Sample\_07

# ちゃぷたー１１　スターを表示しよう

レシピ

　１．　modelData/star/star.cmoを表示するStarクラスを作成する。

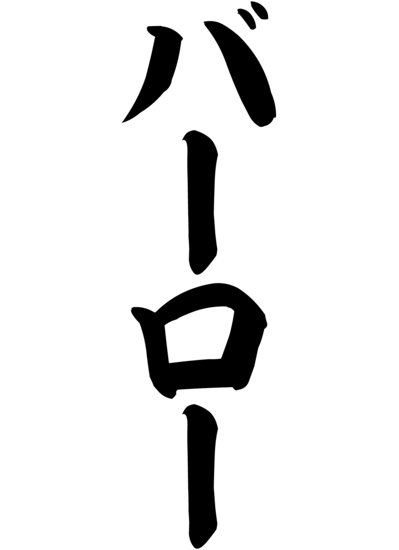
　２．　GameクラスのコンストラクタでStarクラスのインスタンスを５個作成する。

　３．　２で作ったインスタンスを座標を設定して、重ならないように表示する。

　４. Gameクラスのデストラクタで２で作ったインスタンスを破棄する。

　５．　タイトル画面と行き来して、問題ないか確認する！

複数のインスタンスを破棄する方法をしっかり考えるんだ！真実は１つじゃないぞ！



完成動画



参考資料

　１．　やさしいC++ Lesson9 配列

　２．　ゲームプログラミングの教材.pdfのLesson 16\_6

# ちゃぷたー１２　スターを回してみよう

レシピ

　１．　StarクラスにCQuaternion型のメンバ変数を追加しよう。

　２．　StarクラスのUpdate関数の中で、１で追加したクォータニオンに、毎フレームちょっとづつ回転を加算しよう。

　３．　CSkinModelRenderに回転クォータニオンを伝えて、モデルを回転させよう。

クォータニオンは回転を表現する数。クォータニオン同士の掛け算をすることで、回転を加算することができる。



完成動画



参考資料

　tkEngine2/Ｓａｍｐｌｅ/Sample\_20

# ちゃぷたー１３　スターをゲットしてみよう

レシピ

　１．　プレイヤーと星の距離を計算しよう。

　２．　星との距離が一定値以下だったら、星のインスタンスをDeleteGOで破棄。

他のゲームオブジェクトの相互参照！ちょっと難しいけど、ここをマスターしたら、簡単なゲームなら作れるようになるぞ！がんばるんじゃ！



完成動画



参考資料

　　１．　やさしいC++ Lesson9 配列

　　２．　ゲームプログラミングの教材.pdfのLesson\_06\_04

# ちゃぷたー１５　星を全部取れたらゲームクリア演出を行おう

レシピ

　１．　ゲットできた星の数をカウントしよう。

　２．　星の数が５になったら、sprite/GAMECLEAR.ddsを表示しよう。

　３．　GAMECLEAR.ddsを表示して、60フレーム経過したら、タイトル画面に戻ろう。

完成動画



参考資料

　なし