Unityはじめるよ
～ゲームの作り方～

統合開発環境を内蔵したゲームエンジン
http://japan.unity3d.com/

※いろんな職業の方が見る資料なので説明を簡単にしてある部分があります。正確には本来の意味と違いますが上記理由のためです。ご了承ください。
この資料内の一部の画像、一部の文章はUnity公式サイトから引用しています。

ゲームの作り方

今回はUnityを使ったスマホ向けゲームの作り方を紹介します。
※企画や仕様については出来上がってる前提の話です。

・今回作るゲームの企画
■タイトル
「穴に入れたい・・・玉を」

■内容
スマホの傾きセンサーを利用して玉を転がして、
障害物を避けながらゴールの穴に入れるまでのタイムを競うゲーム。

■ターゲット
隙間時間がちょいちょいある人

仕様

■画面構成
タイトル画面
ゲーム画面
結果画面

■タイトル画面
・「タイトル」表示、「ハイスコア」表示、「STARTボタン」を配置
・タップでゲーム画面へ

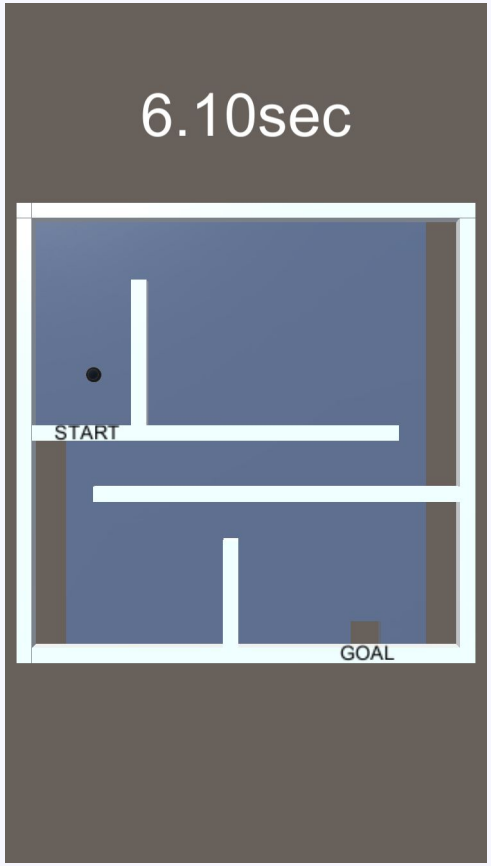
■ゲーム画面
・端末を傾けて玉を転がす
・スタート地点からゴールまでのタイムを競う
・ゴール以外の穴に落ちるとゲームオーバー

■結果画面
・「スコア」表示、「RETRY戻るボタン」配置、「MENUボタン」配置

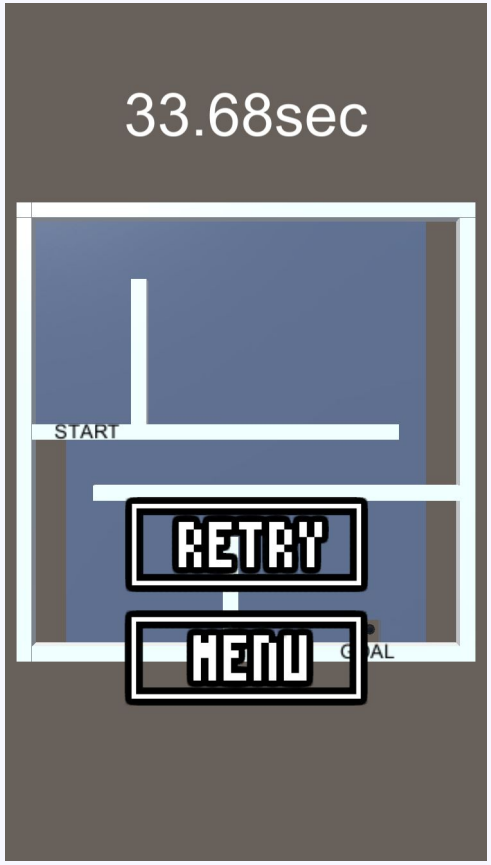
画面構成

タイトル画面

ゲーム画面

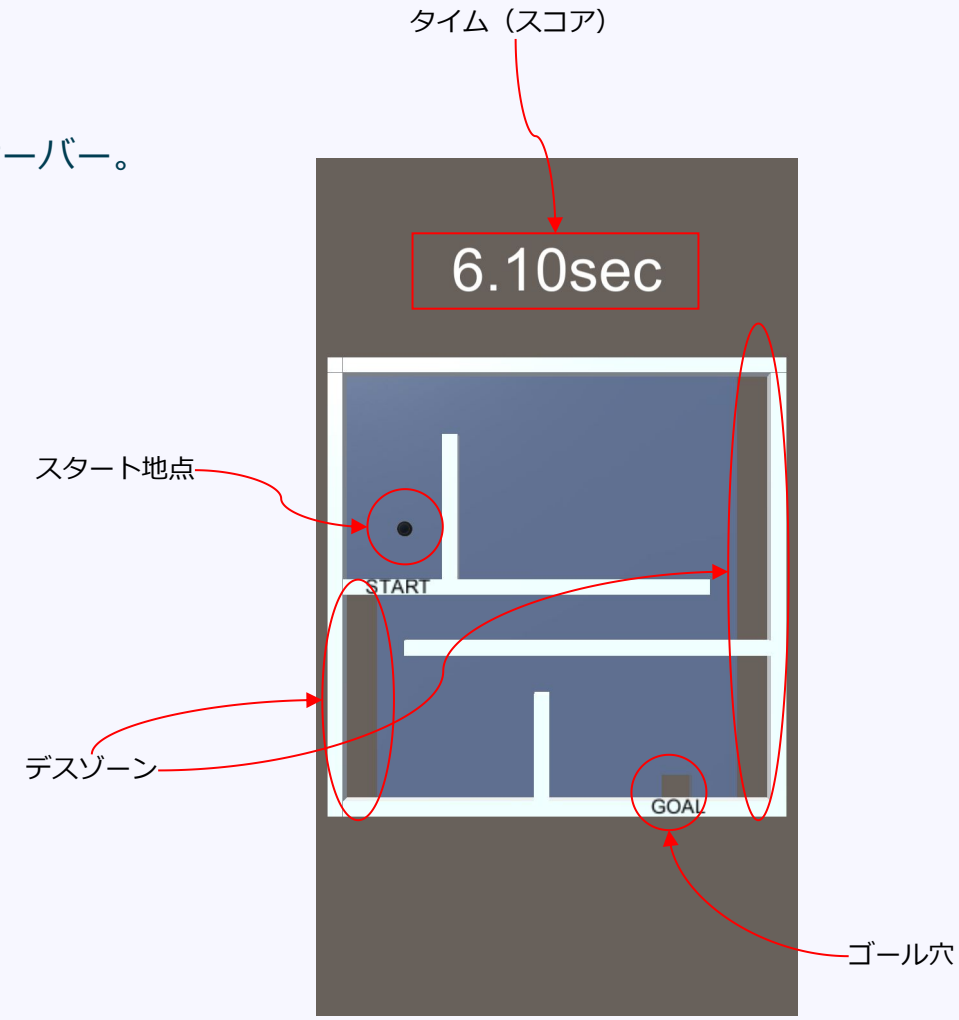


結果画面

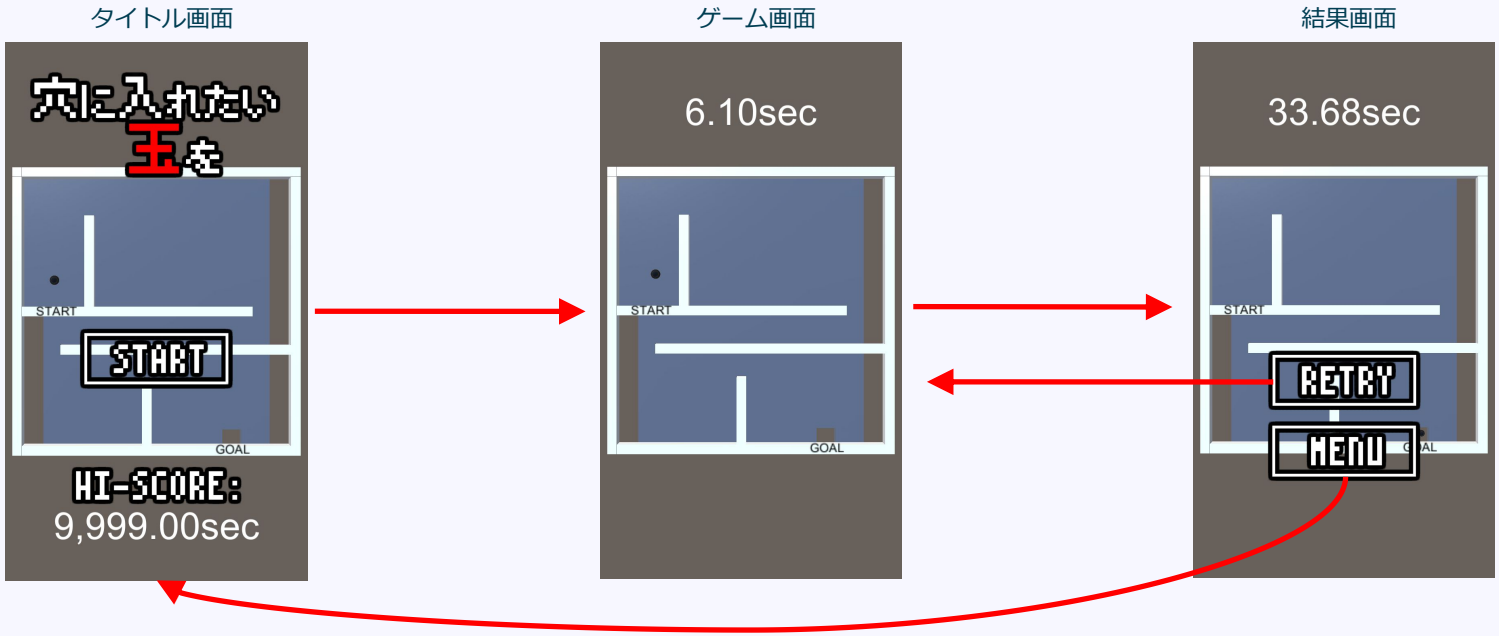


ゲーム概要

端末を傾けて玉を転がし、
スタート地点からゴール穴までのタイム競う。
デスゾーンに玉が落ちるとその時点でゲームオーバー。



画面遷移図



ゲーム作成手順（実演）

サンプルプロジェクト一式
http://monolizm.com/sab/src/PutTheBall.zip

1、プロジェクト作成（フォルダ階層のルール化）

2、プラットフォーム毎の基本情報入力

3、Unity側でのフォルダ構成決め

4、骨格作成
画面作成
ゲームで使用する画面の作成
中身は仮でも、骨組みだけは用意しておく
・タイトル画面
・ゲーム画面
・結果画面
部品作成
ステージ
ボール

5、スクリプトの作成

6、ゲームオブジェクトにスクリプトを割り当てる

1、プロジェクト作成

内容

・Unityプロジェクト
・iOSプロジェクト
・Androidプロジェクト
が収まるフォルダを用意し、
その上でプロジェクトの作成を行う。
フォルダ構成をルール化しておくことで、
プロジェクトの管理がしやすくなるメリットがある。

手順

・「PutTheBall」フォルダを作成。
・その中に「Unity」「iOS」「Android」フォルダを作成。
・Unityを起動し、NewProjectでプロジェクトを作成。
・Project nameを「PutTheBall」に。
・Locationを「PutTheBall/Unity」に。

2、iOS用の設定、Android用の設定

内容

プラットフォームごとの設定を行う。
最初に設定しとくことで、プラットフォーム切り替え時に生まれる可能性のあるバグを、起きにくくする。

手順

・iOSの設定

・File > Project Settings でBuild Settingウィンドウを表示
iOS > Switch PlatofromでiOSを選択しておく
（このタイミングで行わなくてもいいけど、やっとくと一手間少ない）
・iOS > Player Settings でインスペクタビューにiOSの設定項目が現れる
・Company Name > 適当に
・Product Name > 「PutTheBall」
・Default Icon > アイコン選択
・Resolution and Presentationの項目
Orientation > Default Orientation はPortraitにして、今回は縦画面固定のゲームとする
・Other Settingの項目
Identification > Bundle Identifier 事前にiOS Dev Centerで用意した値。

・Androidの設定
  ・インスペクタビュー内上部の、OSアイコンが並んだタブからAndroidを選択
    ※iOS側で設定した項目が反映されているのでPublishing Settingsのみの設定でOK
  ・Publishing Settings > Keystore > Create New Keystore
  ・Brows Keystoreボタンを押し「PutTheBall」という名前でSave（保存場所はデフォルトでOK）
    ※このファイルは非常に重要なので絶対に削除しないこと
    （ストアにアップするときに必要。バージョンアップ時も利用する）
  ・Keystore password 適当に「PutTheBall」
  ・Confirm password 上記と同様のパスワードを打ち込む
  ・Key > Alias > create new key　この項目の詳細は今回は紹介しません
      Alias 「PutTheBall」
      Password 「PutTheBall」
      Confirm 「PutTheBall」
      First and Last Name 名前を入力
      Organizational Unit 会社名を入力
  ・作成後 Alias > puttheball　を選択し Passwordに先ほど設定した「PutTheBall」を入力

・設定を保存
      File > SaveProject

3、Unity側でのフォルダ構成決め

内容

事前にフォルダ構成をルール化して整理しておく。
プロジェクトが大きくなった際の混乱を抑えるメリットがある。

手順

今回はプロジェクトが小さいので、
下記のフォルダを作成するだけ。

- textures
- prefabs
- scripts
- sounds
- materials

４、骨格作成

内容

ゲームで使用する画面の作成。
今回は軽いゲームなので、この時点で全部品を作成しておくが、
規模の大きなゲームは、中身は仮でもよい。
ただし、画面遷移に必要な骨組みだけは用意しておく。

手順

・必要なリソースの読み込み。（今回はres.pngのみ）
・uGUIを利用して、各画面の作成
・ステージの作成
・ゴールとデスゾーンの作成
・玉の作成

5、スクリプトの作成

内容

ゲームを制御するためのスクリプトの作成。

手順

・プロジェクトビューから後述の８種類のスクリプトを作成
※ソースの中身はサンプルプロジェクトをご確認ください

スクリプトの種類と内容

ゲーム全体をコントロールする**MainManager**クラス
　アプリ全体の初期化処理と、画面遷移制御を行う。
　画面遷移は全てこのクラスを通して行うことで、各画面の独立性が高まり、
　バグの発生の減少と、可読性が向上する。

各画面をコントロールするクラス群
　各画面共通処理用基底**BaseManager**クラス
　　各画面で共通となる処理はこのクラスに記述し、各画面クラスはこのクラスを継承する形で実装する。

　タイトル画面**TitleManager**クラス
　　タイトル画面の制御を行う。
　　主な処理はSTARTボタンの処理と、ハイスコア表示。

　ゲーム画面**GameManager**クラス
　　ゲーム画面の性をお行う。
　　主な処理は、タイム管理、ゴール判定、死亡判定。

　結果画面**ResultManager**クラス
　　結果画面の制御を行う。
　　主な処理は、スコア表示と、RETRYボタン制御、MENUボタン制御。

部品を制御するクラス群
　玉制御**BallController**クラス
　　加速度センサーからの値を見て、玉のトランスフォームに力を加える。

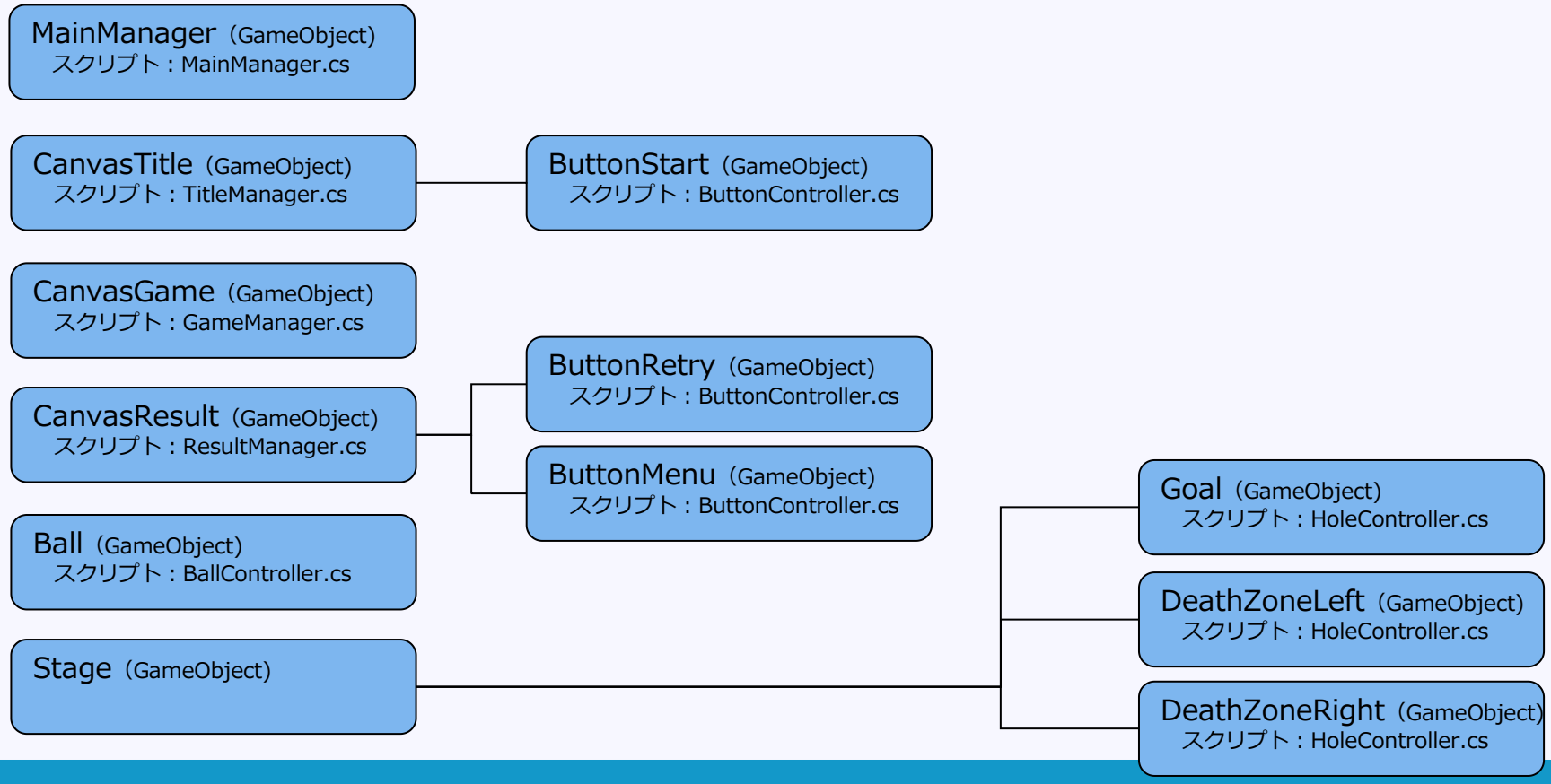
　穴制御**HoleController**クラス
　　玉が穴に入ったかの判定を行う。

　ボタン制御**ButtonController**クラス
　　ボタンがタップされたかどうかの判定を行う。

6、ゲームオブジェクトにスクリプトを割り当てる

内容

作成したスクリプトをGameObjectに割り当てる。
ソースコードをガリガリ書くのは、この後の作業となることが多い。



実演での説明が多いので、この資料だけでは伝わりにくいですが、Unityでのゲーム開発の作業の流れはこんな感じです。

その場で考えるような作業はしないで、事前にルールを決めて整理しておくことが大事です。
Unityは部品（ゲームオブジェクトとコンポーネント）の、独立性が高い仕組みになっているので、ちゃんと整理することを心がけていれば、非常にわかりやすい構成を保ったまま開発を進められます。
それゆえに、再利用可能な部品も作りやすいので、特定のゲームに依存しない共通な処理は、できるだけPrefab化しやすいフォルダ構成にしておくと良いと思います。（フォルダのコピーで他のゲームに持っていけるイメージ）

サンプルプロジェクト一式
http://monolizm.com/sab/src/PutTheBall.zip

ご清聴ありがとうございました