

株式会社アウトスタンディング さいたまゲーむす 勉強会資料

勉強会情報

<https://www.facebook.com/groups/664428513635931/>

アプリ作ってます！

Android

https://play.google.com/store/apps/developer?id=SAITAMA_GAMES

IOS

<https://itunes.apple.com/jp/app/pichannobaningu!/id880087045>

使用する素材

<http://saitamagame.com/study/speedmove/WEB/SpeedMove.zip>

完成版

<https://dl.dropboxusercontent.com/u/75315662/study/0713SpeedMove.zip>

●勉強会で学べること

Unityによるテンプルラン風味のゲーム作成。

Unityでのオブジェクト作成等

Skybox FOGなどを使った背景演出

●作る内容

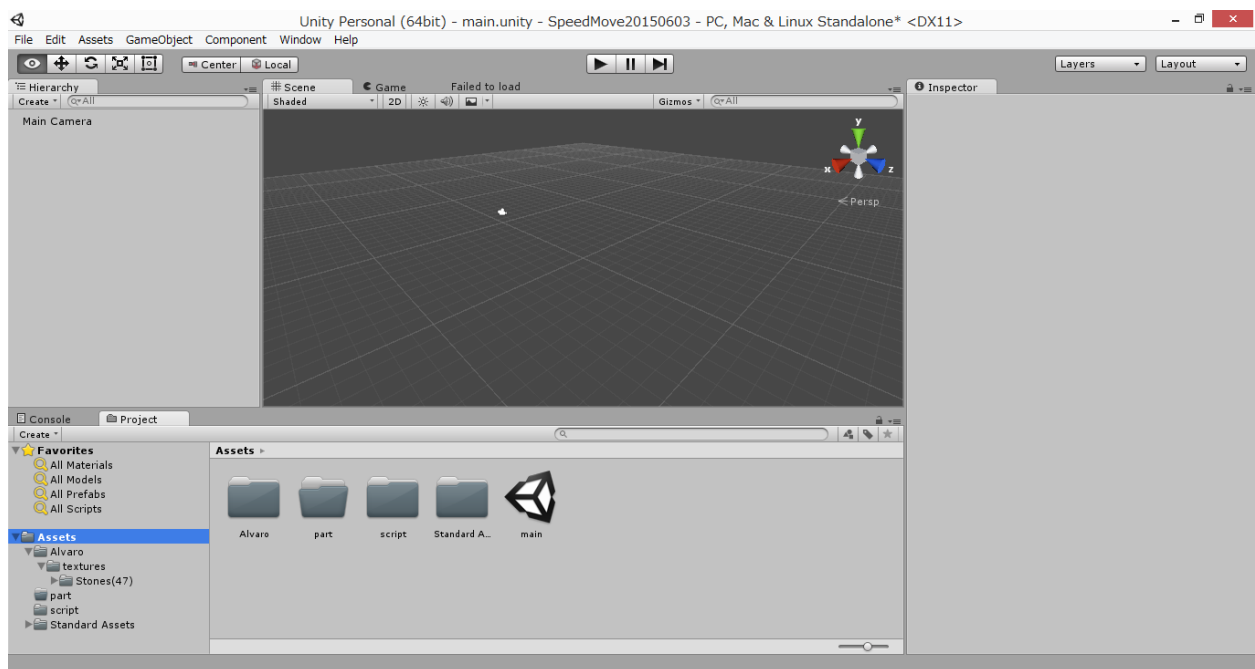
自分は前へどんどん進み、障害物を避けていく、
障害物にあたった時点で終了という内容のゲームを作成します。

●0準備

0-1) 使用する素材SpeedMove.zipをダブルクリックで解凍し、解凍されたものを全て適当なフォルダにいれる

0-2) Assetsフォルダ内のmain.unityをダブルクリック unityを起動する

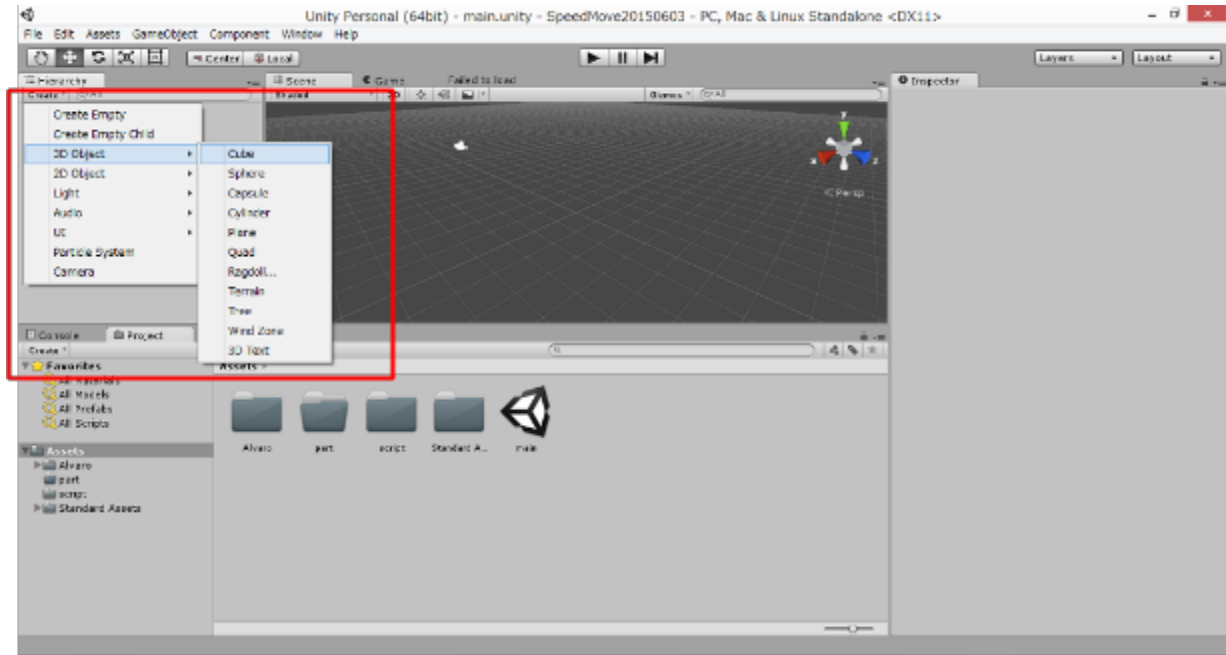
立ち上げ時の画面



● 1 背景の作成

1-1) 床オブジェクトの作成

Hierarchy>Create>3D Object>Cube でキューブを作成

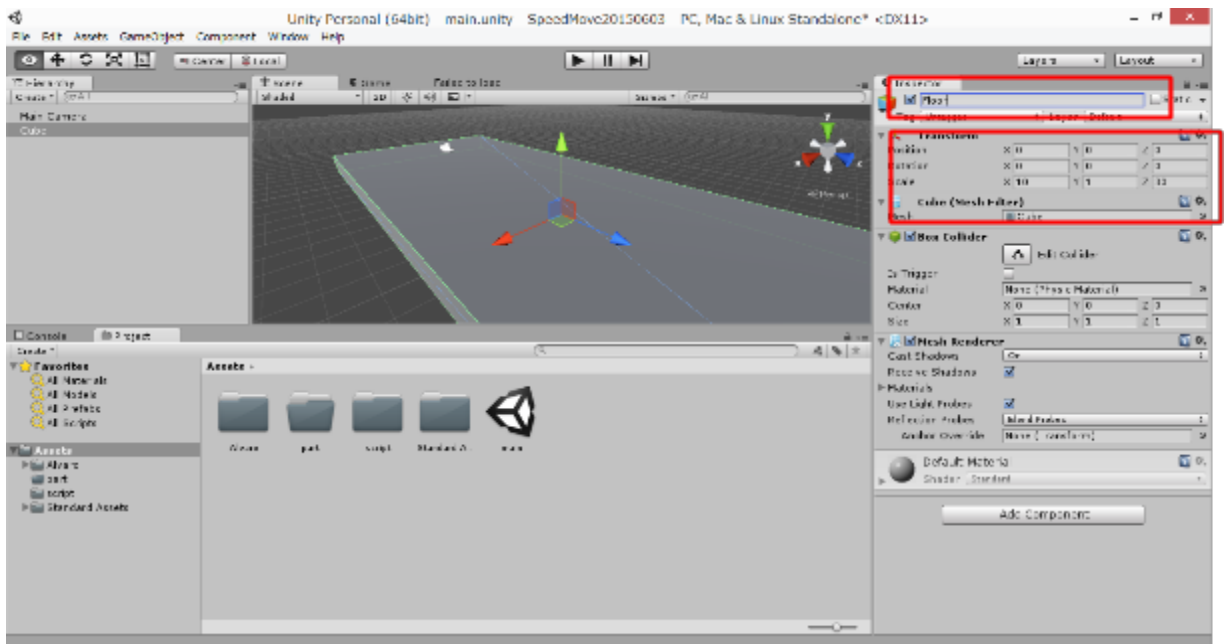


これを拡張して床にします

Name:**Floor**

Transform Position x:0 y:0 z:0

Transform Scale x:10 y:1 z:30



1-2) 同様に左右の壁を作成

Hierarchy>Create>3D Object>Cube でキューブを作成

NAME: **Wall1**

Transform Position x:6 y:1 z:0

Transform Scale x:1 y:5 z:30

Hierarchy>Create>3D Object>Cube でキューブを作成

NAME: **Wall2**

Transform Position x:-6 y:1 z:0

Transform Scale x:1 y:5 z:30

1-3) 障害物作成

Hierarchy>Create>3D Object>Cube でキューブを作成

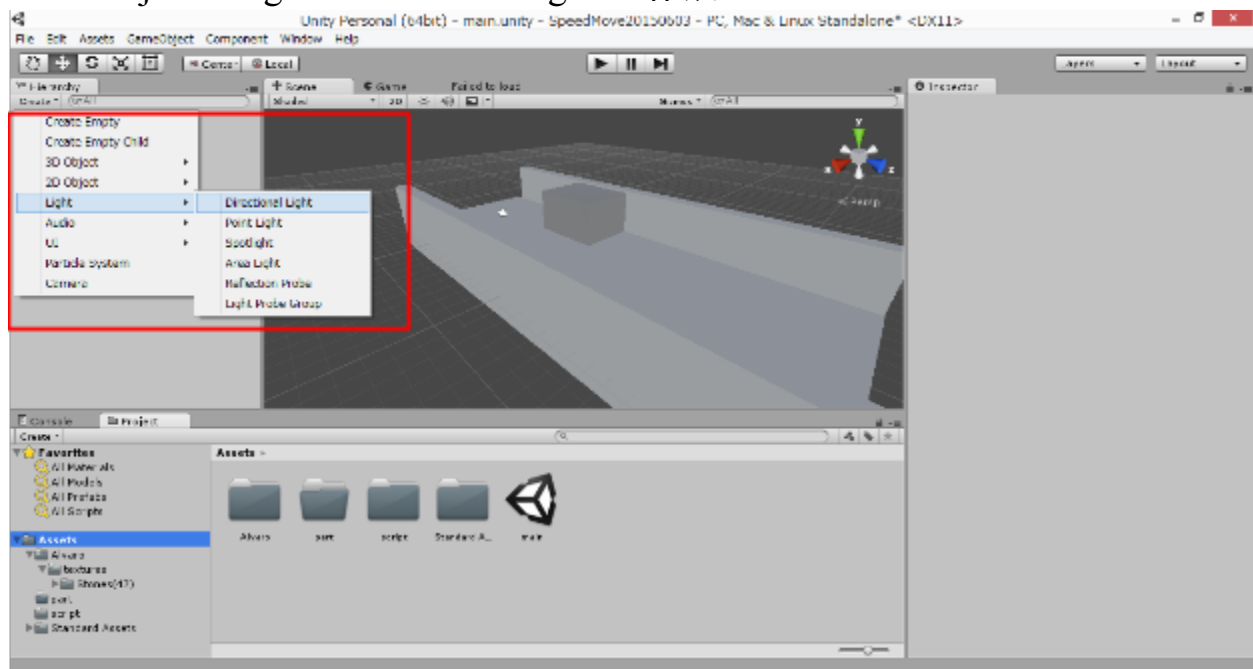
NAME: **Obj1**

Transform Position x:-3 y:1 z:-3

Transform Scale x:4 y:5 z:5

1-4) 明かり作成

GameObject->Light->Directional light で作成



明かりがつきます。

1-5) テクスチャー貼り付け

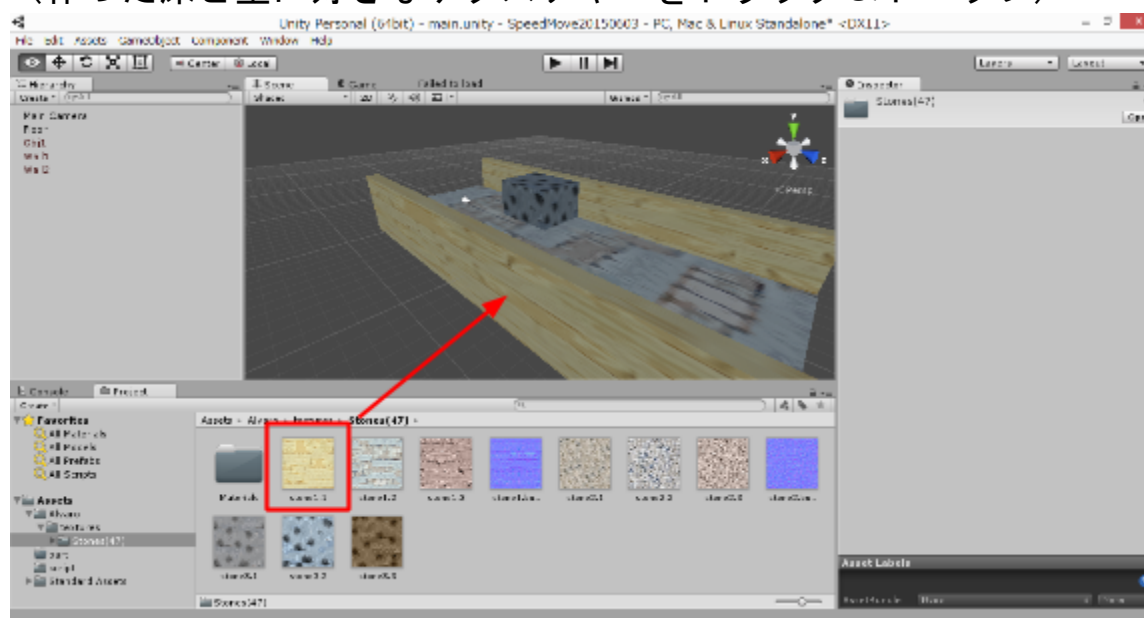
Assets\Alvaro\textures\Stones(47)

に入っているテクスチャーを張り付けます。

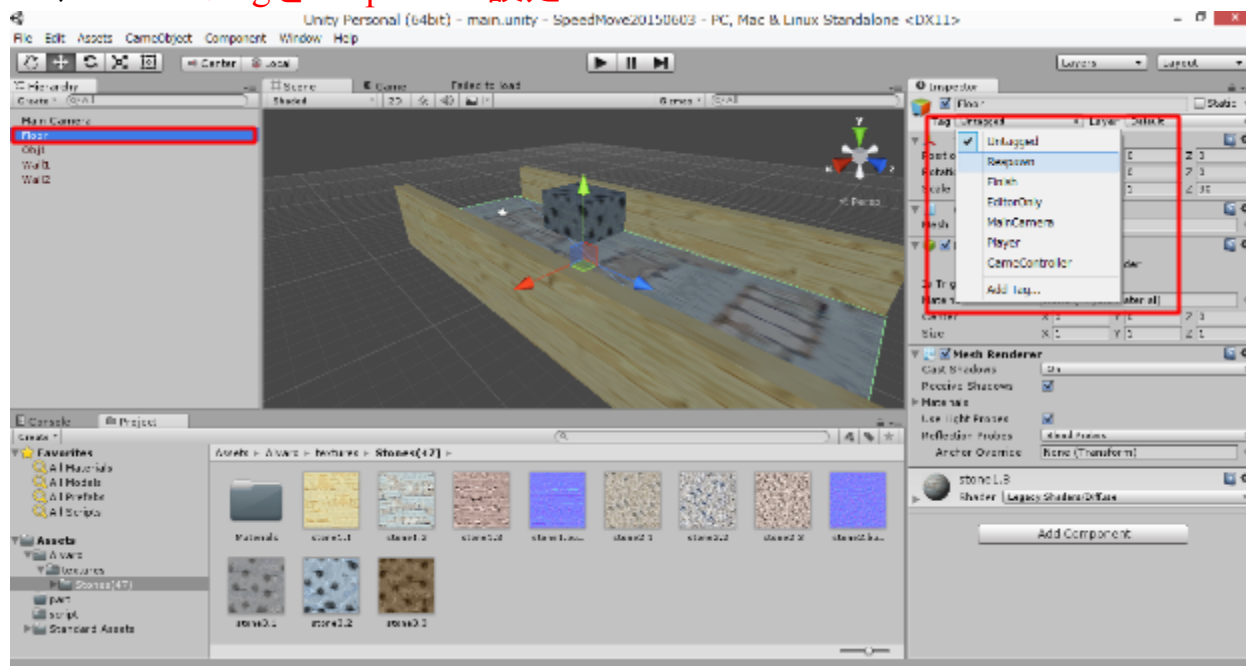
Project→Assets\Alvaro\textures\Stones(47)

の絵のファイル（テクスチャー）を画面上のオブジェクトへドラッグして張り付け

（作った床と壁に好きなテクスチャーをドラッグ&ドロップ）



1-6) FloorのtagをRespawnに設定

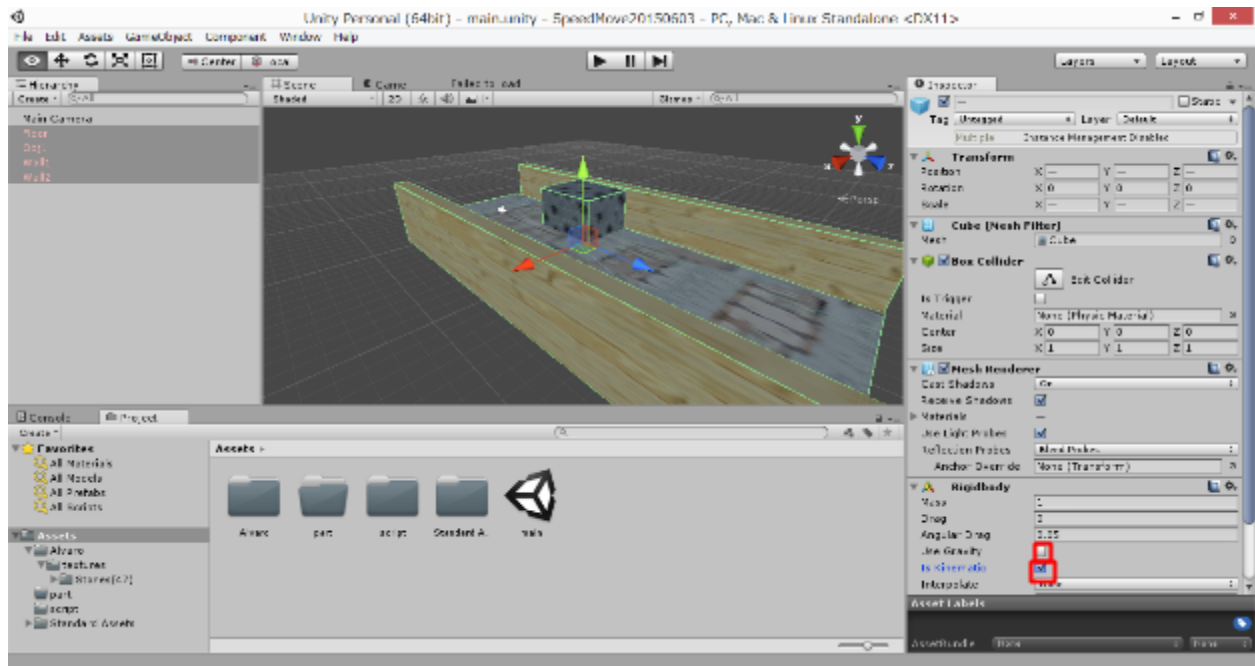
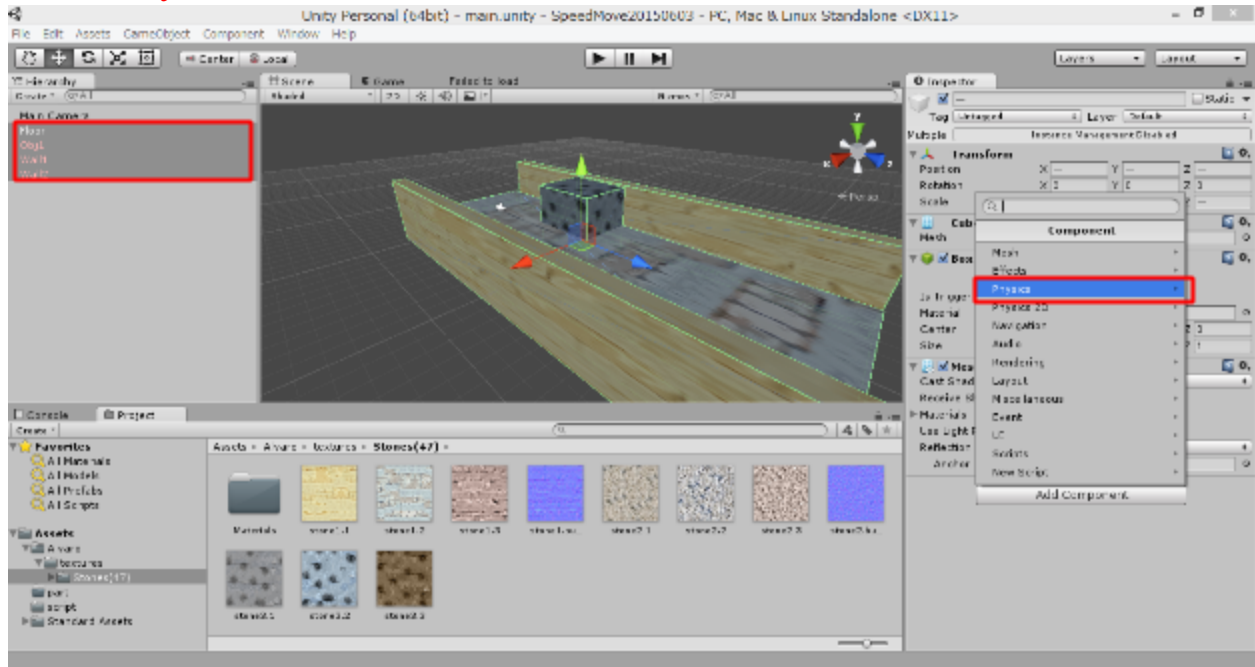


tagで当たり判定の識別をするので、床のtagを変更します

1-7) Rigidbodyの追加

各オブジェクトのInspectorの[AddComponent]->[Physics]->[Rigidbody]で Rigidbodyを追加

UseGravityをオフに IsKinematicをONに設定します。



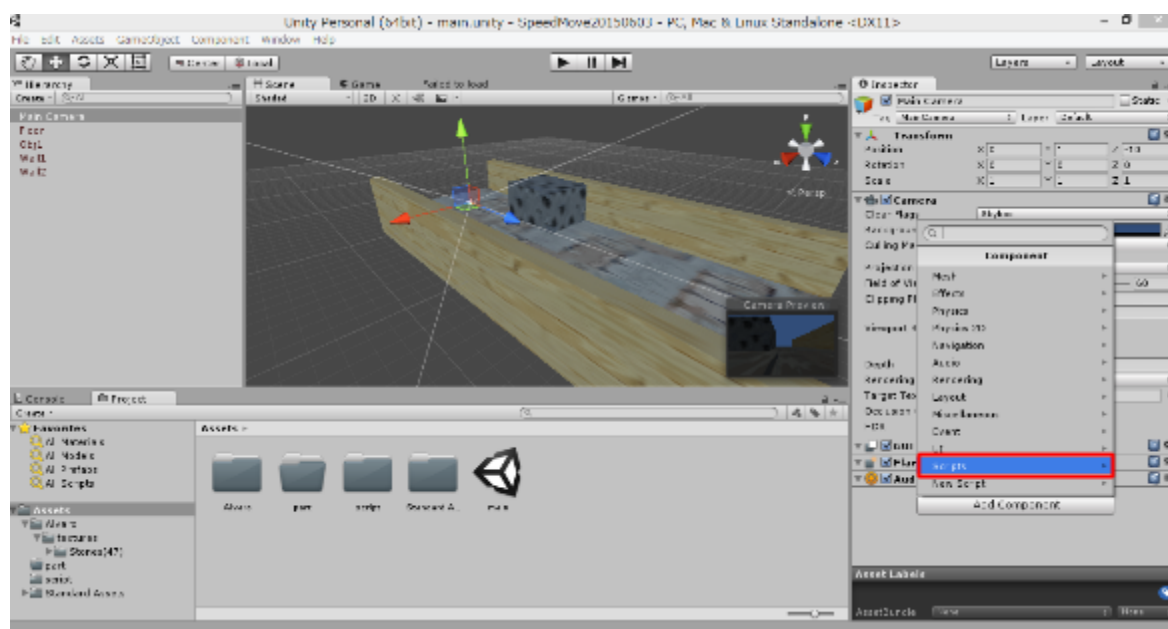
●2 プレイヤー（カメラ）の移動 他カメラ周りの設定

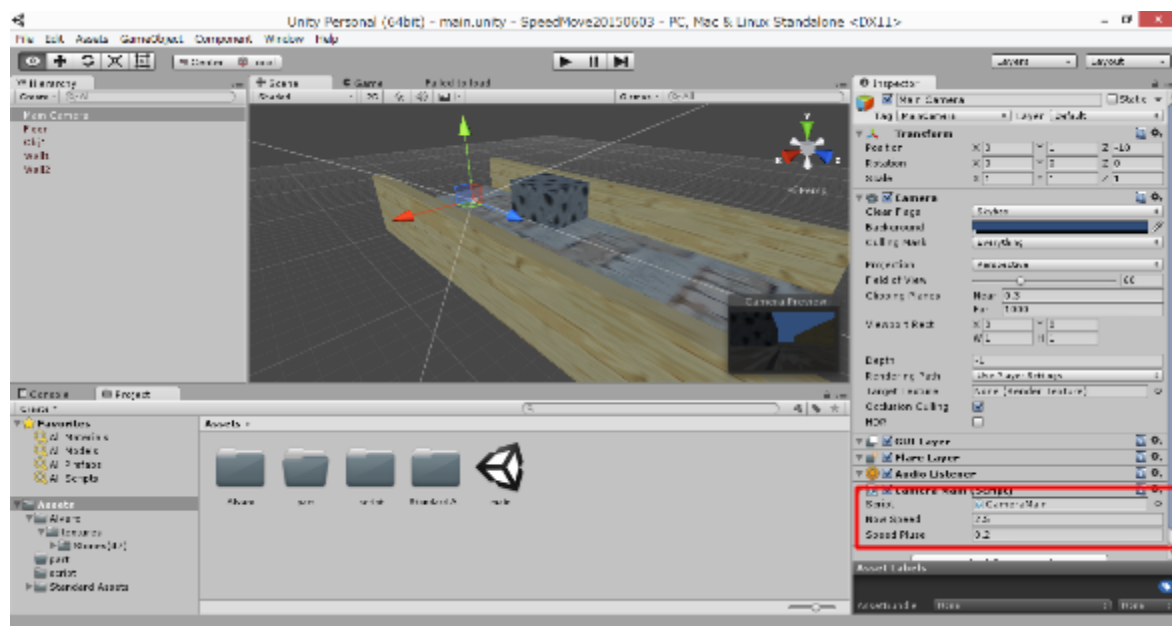
2-1)スクリプトの適応

MainCameraにカメラの動きをするスクリプト「CameraMain.cs」を入れます。左右に動けるようになります。

MainCameraのInspector 内の[AddComponent]>[Scripts]>[CameraMain] で設定します。

NowSpeed SpeedPluseで移動速度の調整が可能です。

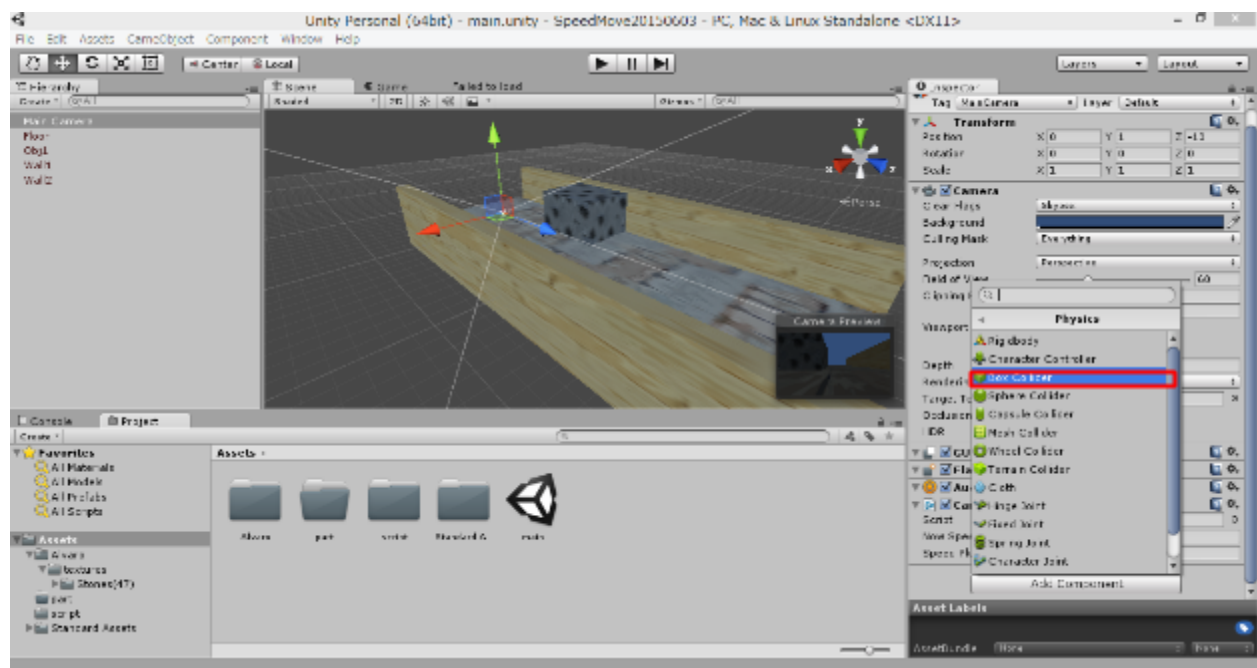


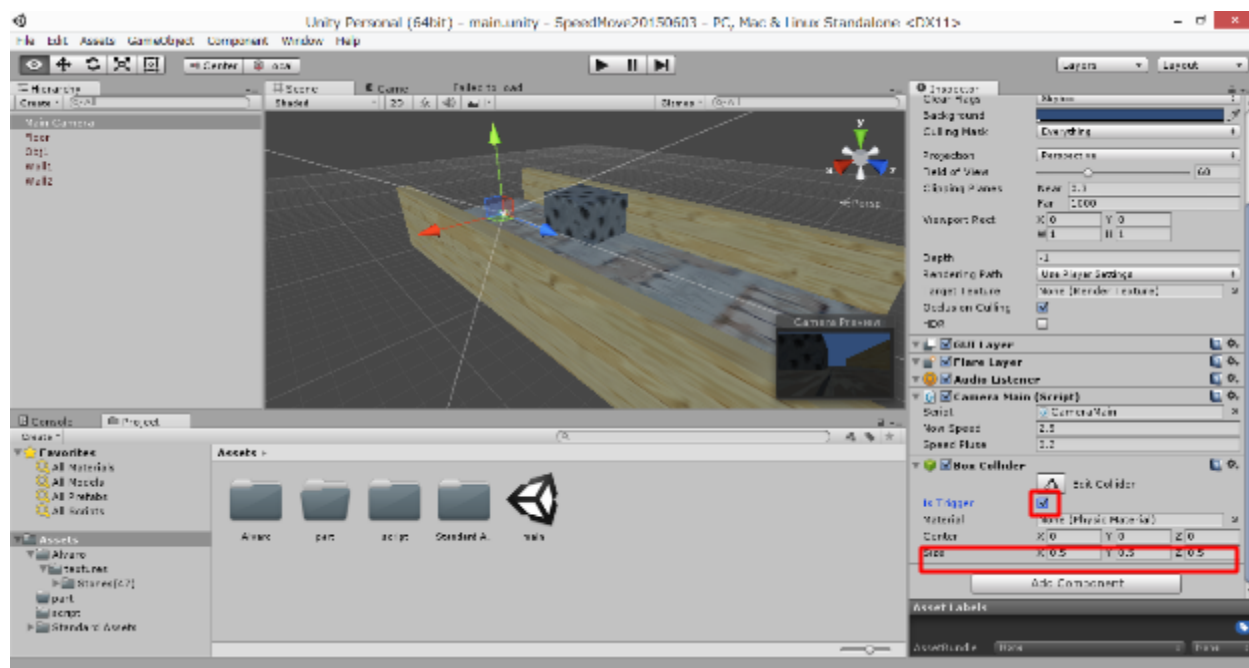


2-2)物理計算系追加

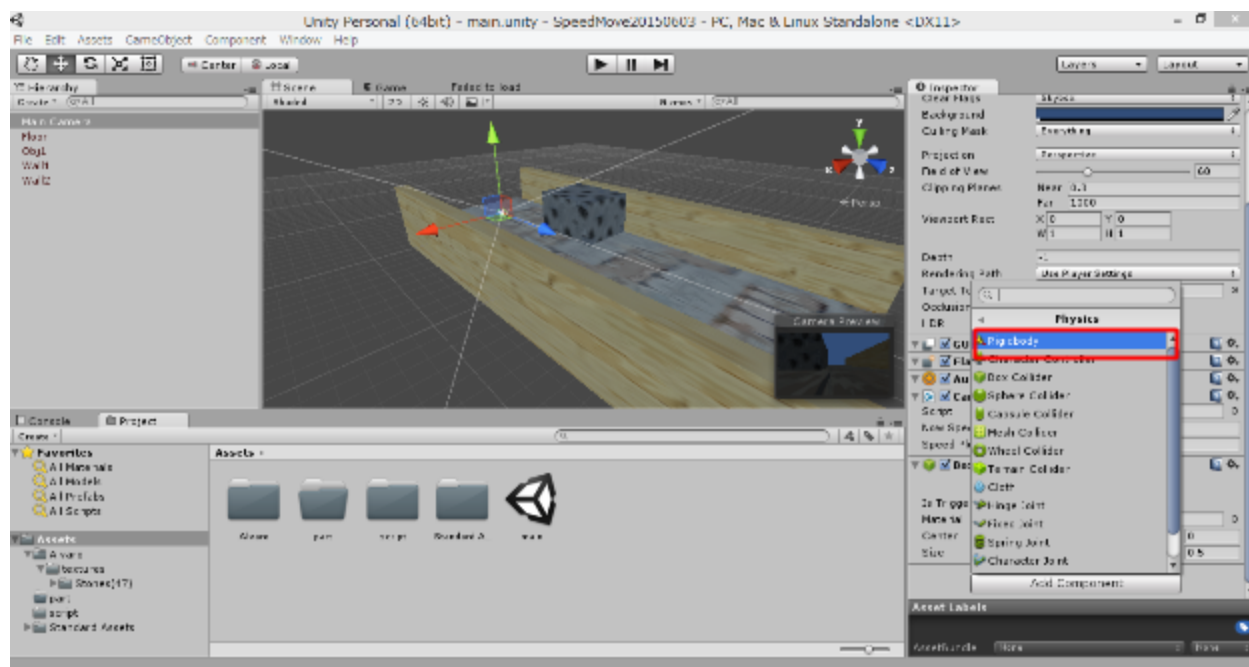
- ・ MainCameraにBoxCollider追加

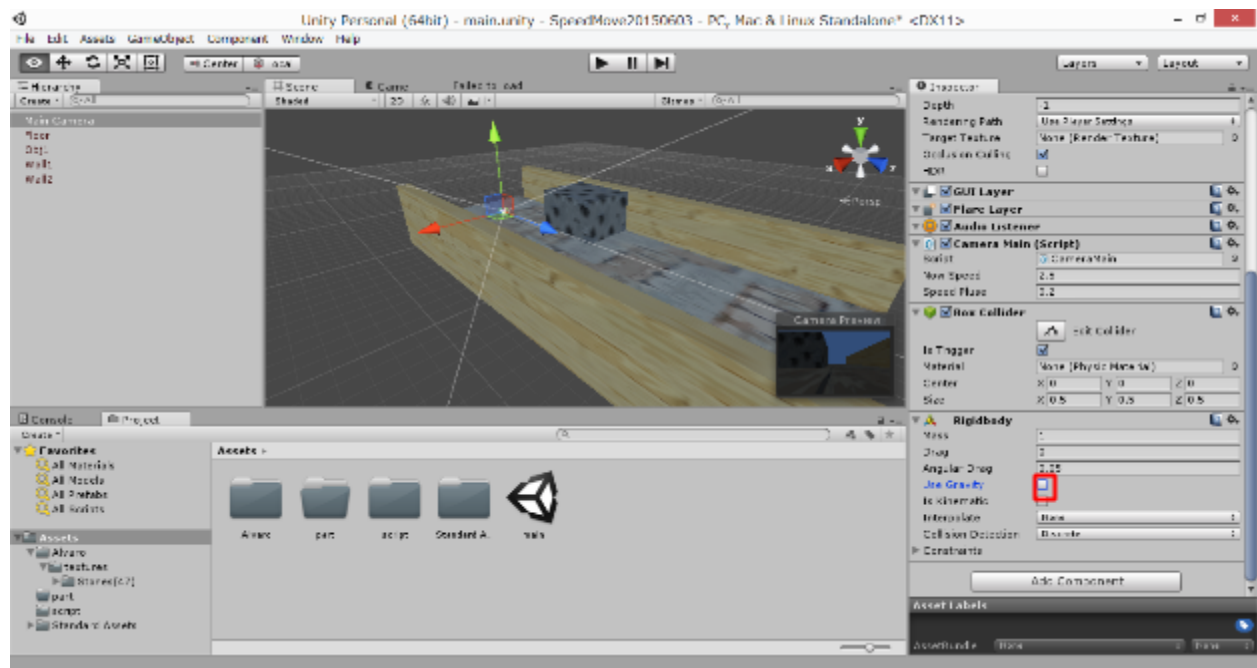
InspectorのAddComponent>Physics>BoxColliderでBox Colliderを追加
大きさを X 0.5 Y 0.5 Z 0.5 に設定 **isTrigger**を**チェック**



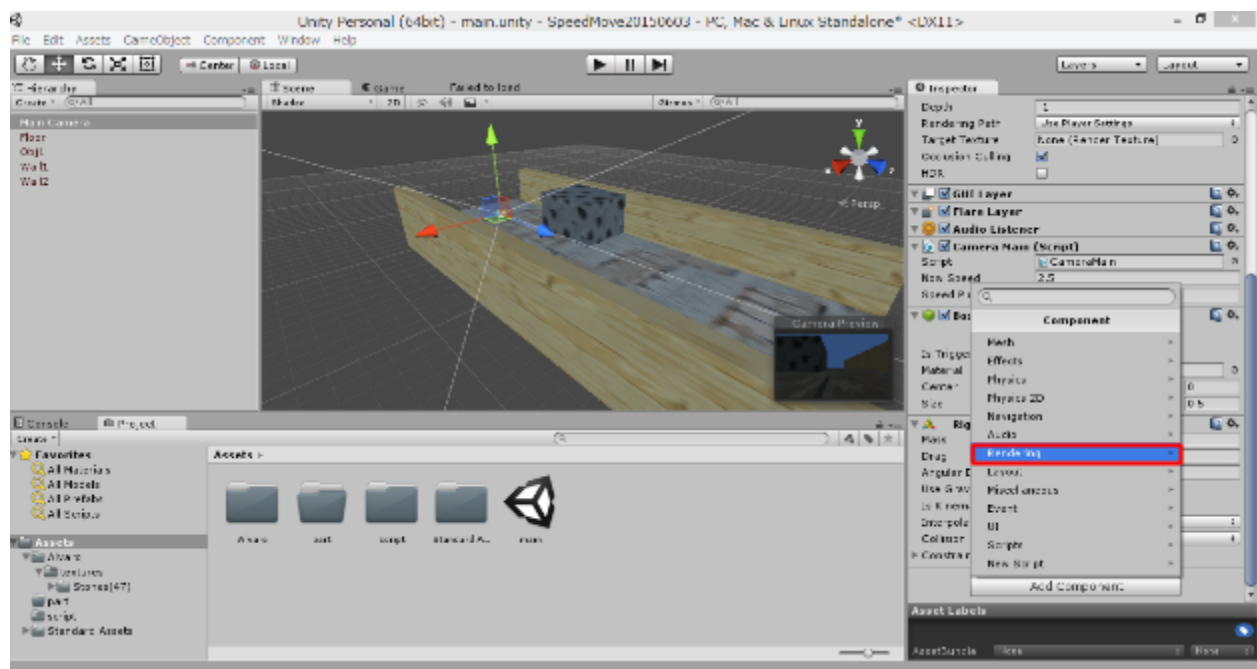


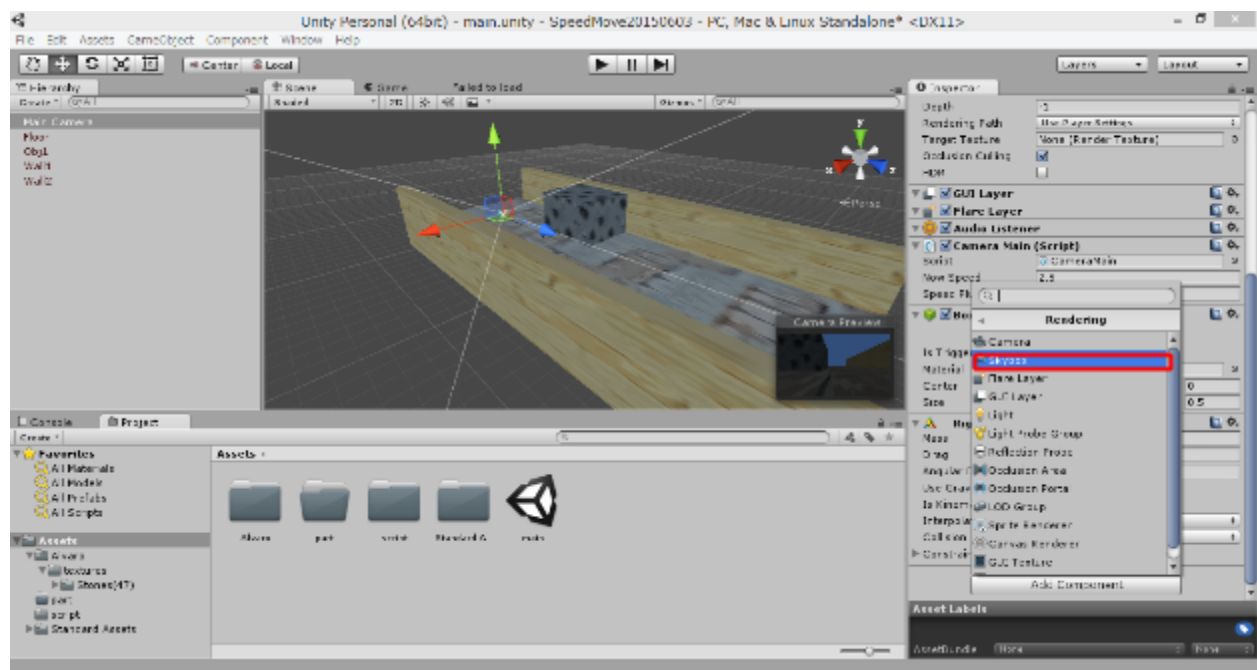
・ Rigidbody 追加
Inspectorの[AddComponent]->[Physics]->[Rigidbody] Use GravityをOFF





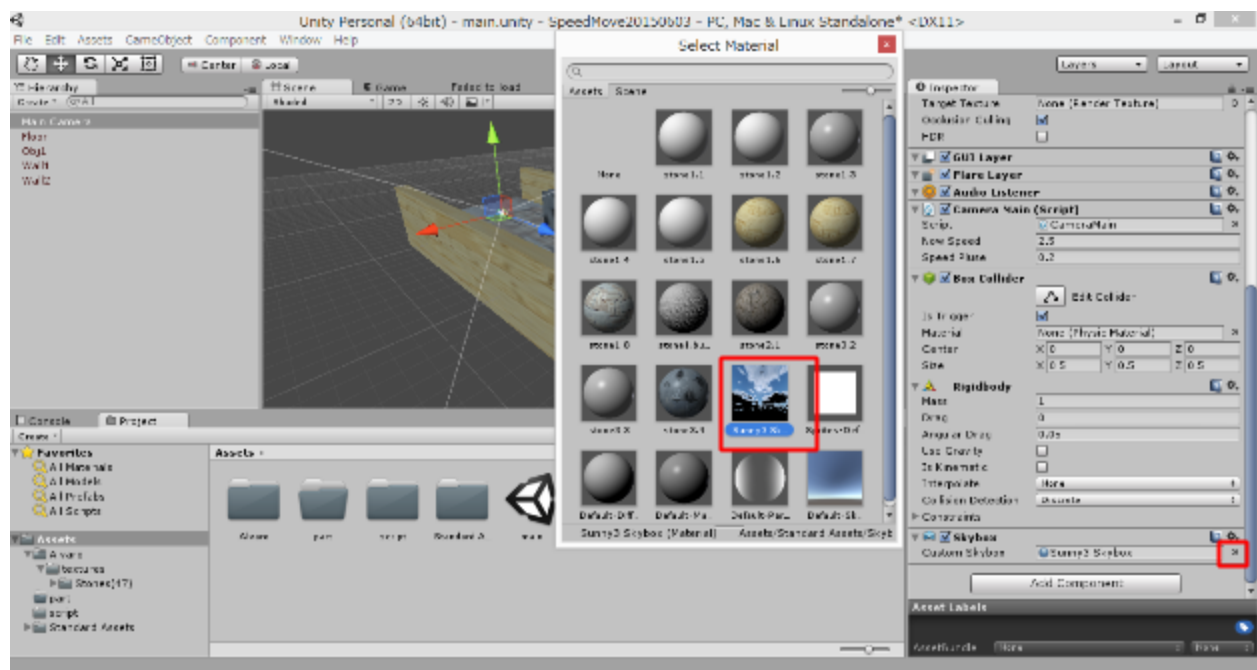
2-3)Inspectorの[AddComponent]->[Rendering]->[Skybox]を選択





2-4)skyboxのテンプレートを設定

Inspectorの Custom Skybox の右側にある丸をクリックして、Select Materialウィンドウを開きます。



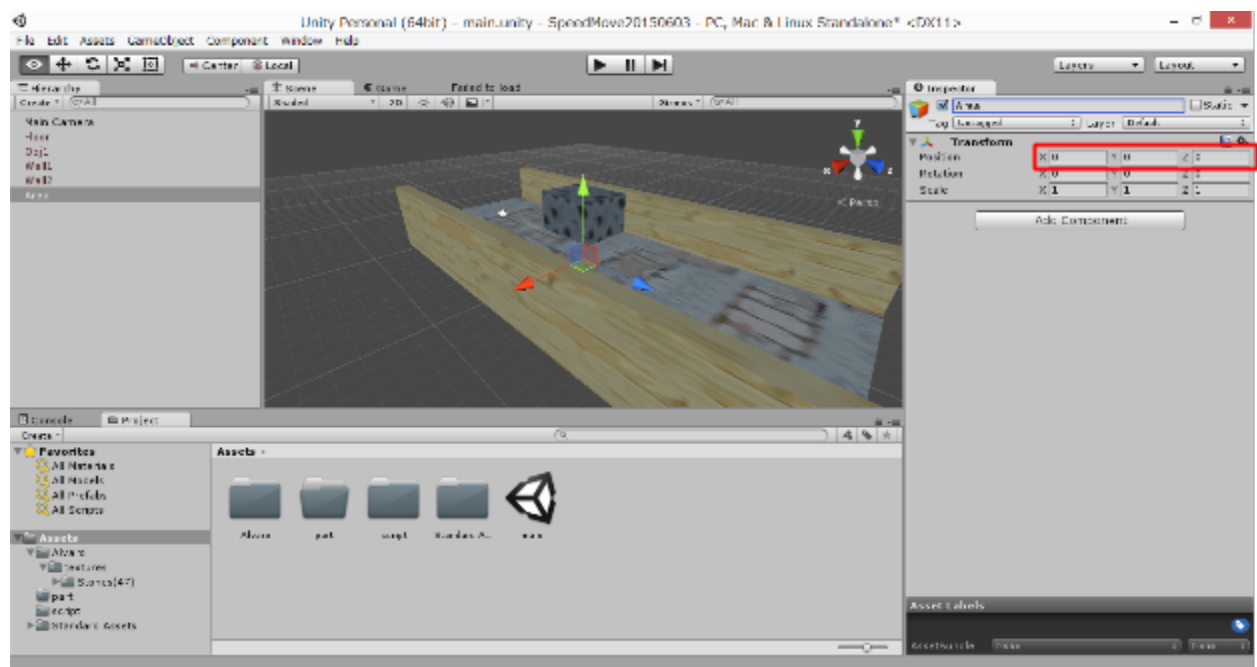
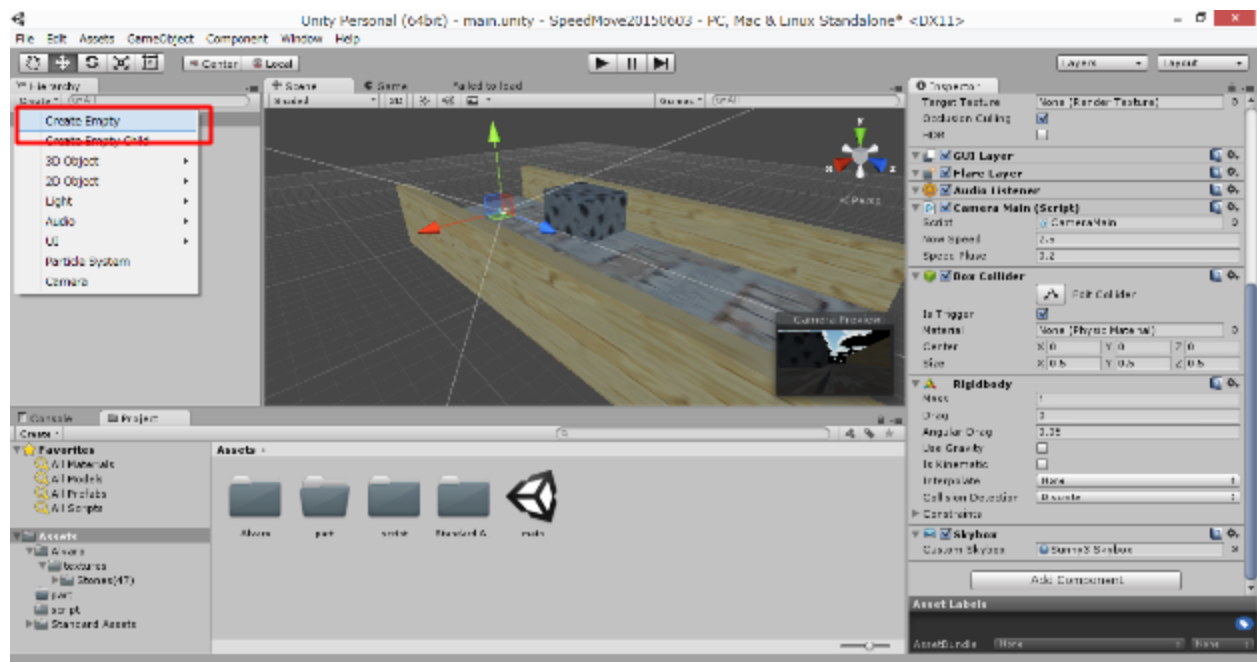
その中から好きな画像をダブルクリックすることで空が適応されます。
今回はSunny3 Skyboxを使用します。

●3ステージ作成など

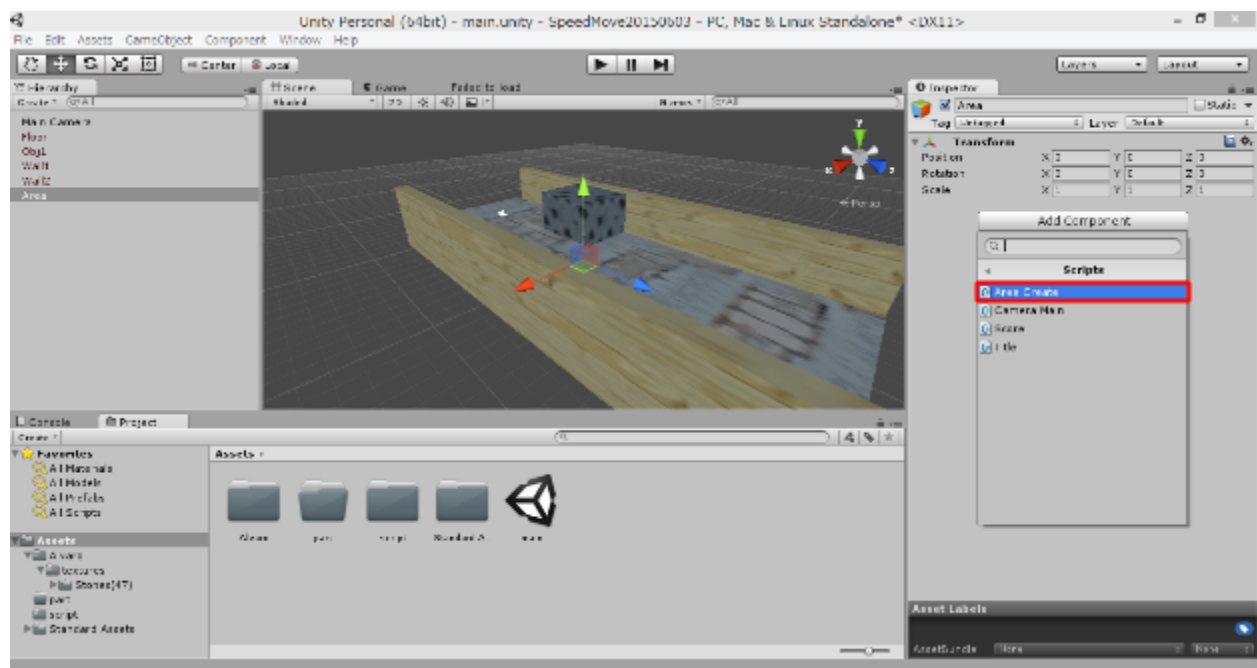
3-1) マップ自動生成システム

Unityメニューの[GameObject]->[CreateEmpty]を選択します。

名前は”Area”にしましょう。
Transform Position x:0 y:0 z:0



AreaのInspector 内のAddComponent>Scripts>AreaCreate で設定します。
設定したマップを自動で生成していきます。



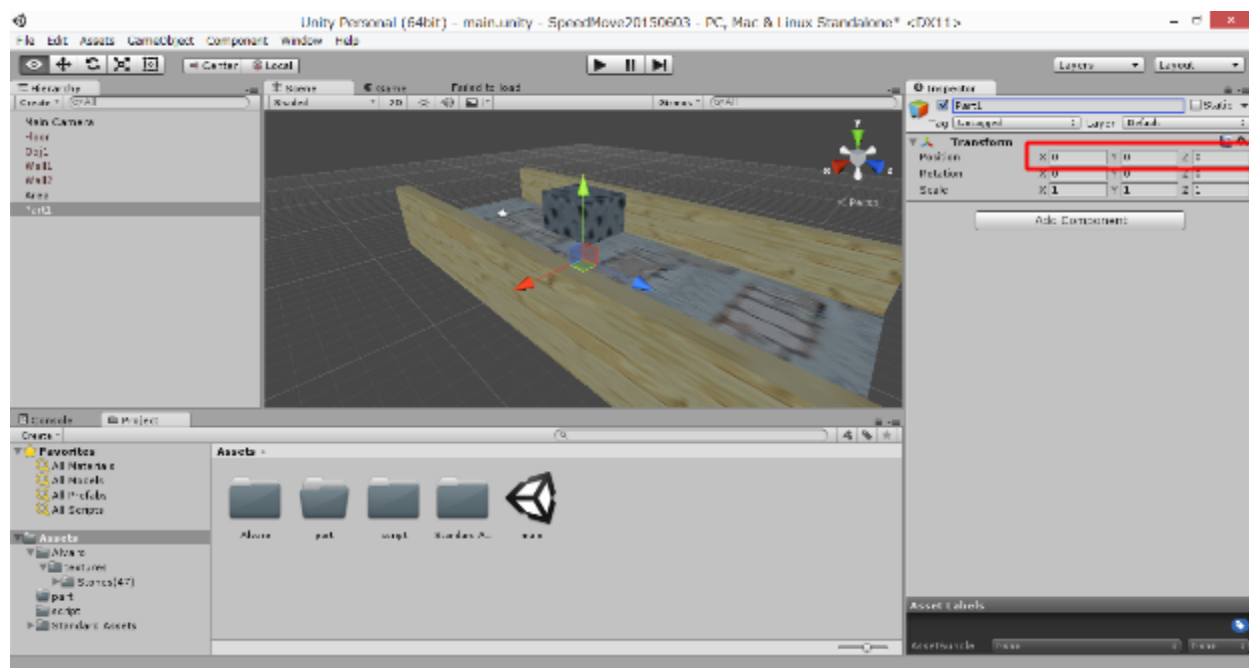
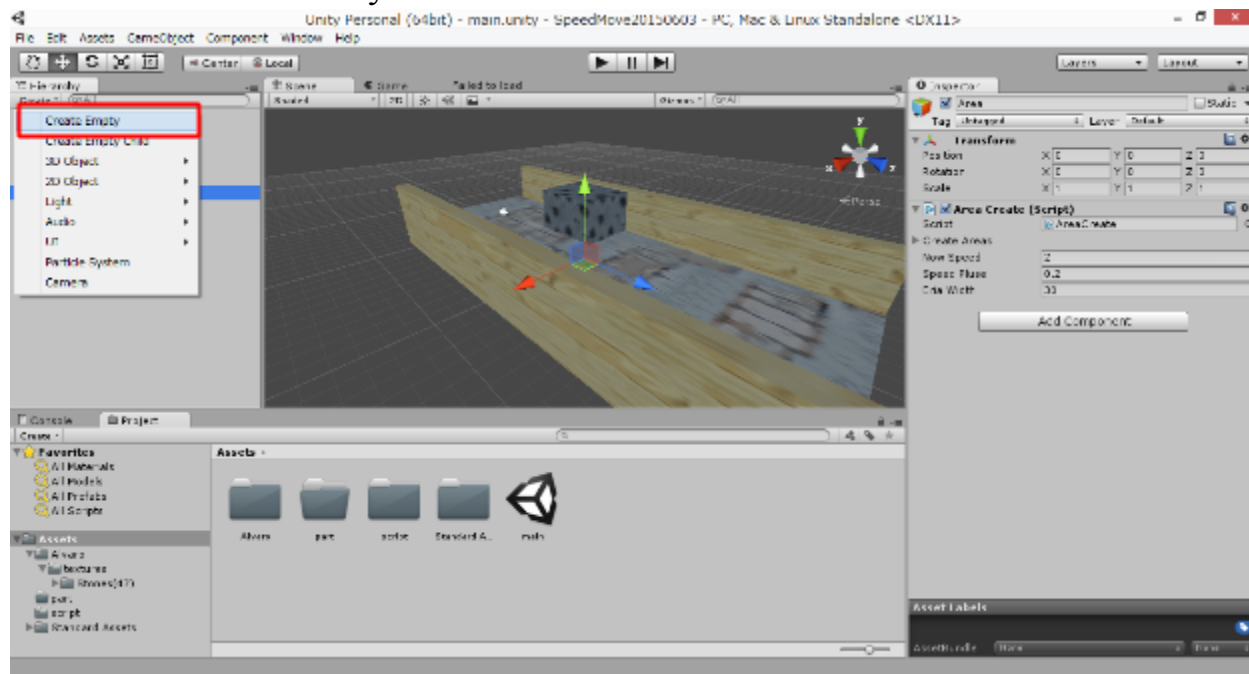
3-2)オブジェクトのプレファブ化
今まで作ったオブジェクトをプレファブ化します。

その前に一まとめにしておきます。

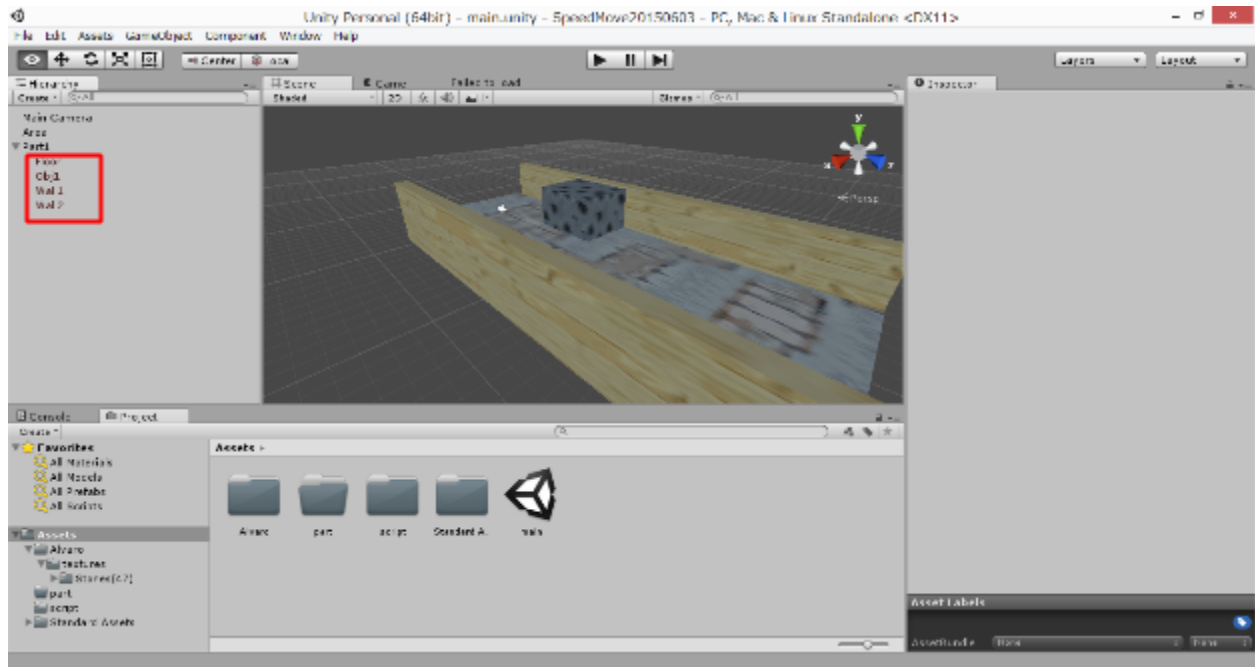
Unityメニューの[GameObject]->[Create Empty]を選択します。

名前は”Part1”にしましょう。

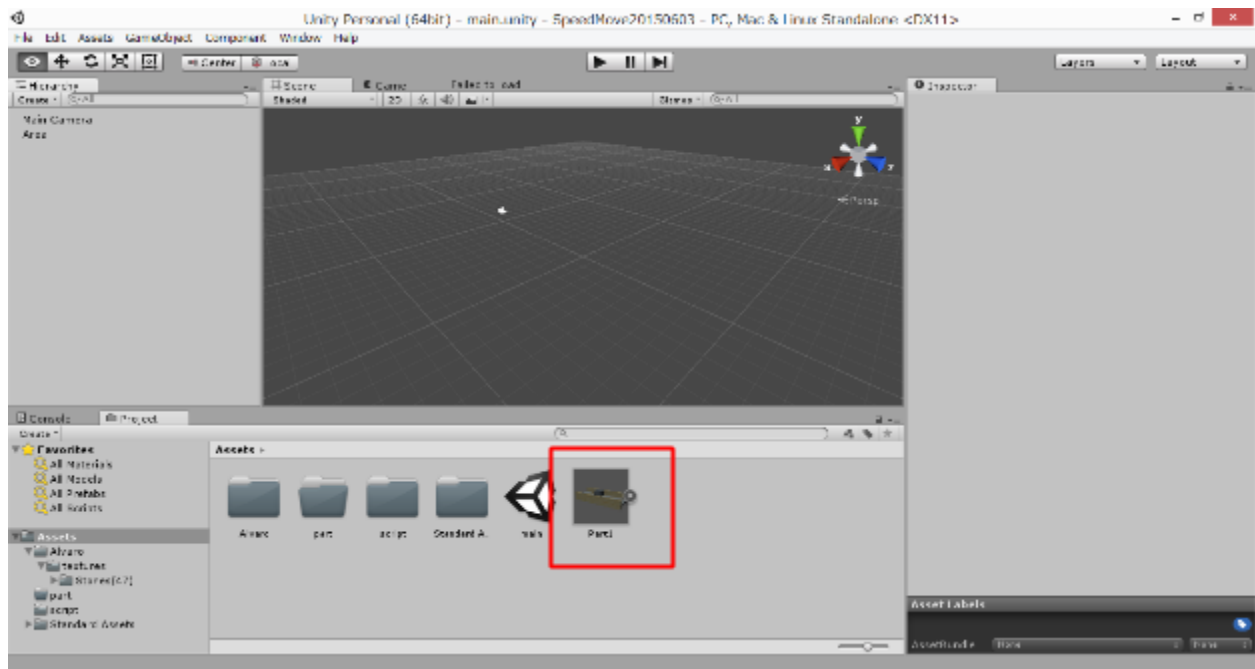
Transform Position x:0 y:0 z:0



そのPart 1に今まで作ったオブジェクトをドラッグして親子関係にします。

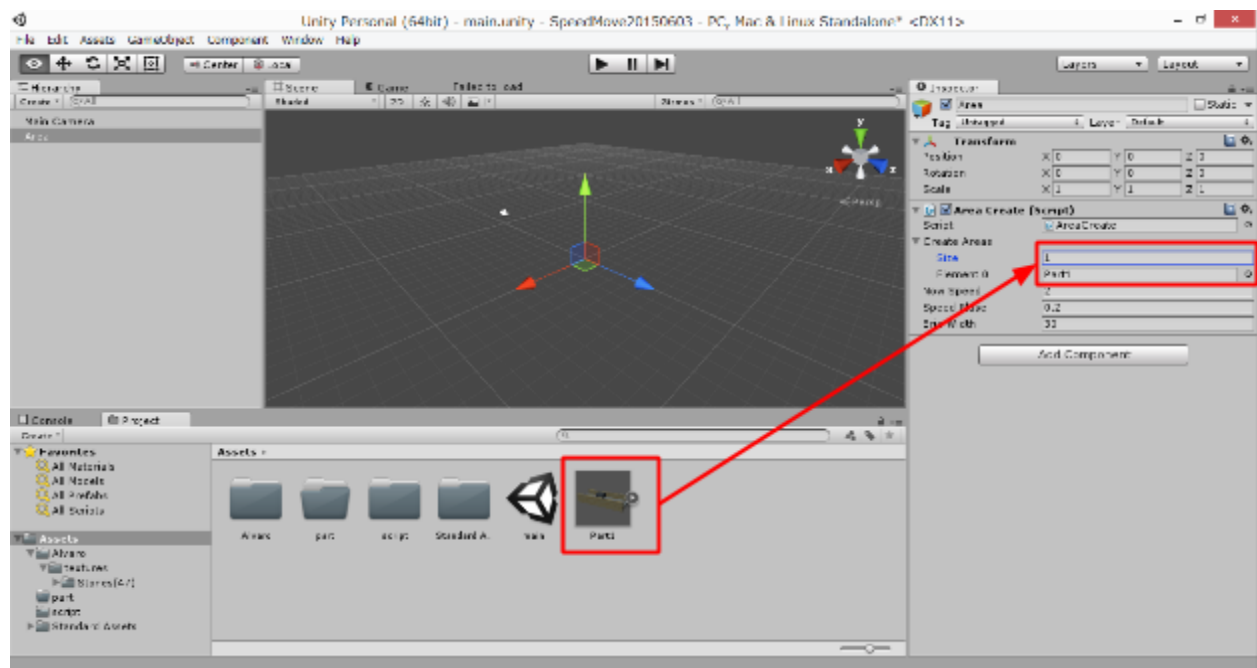
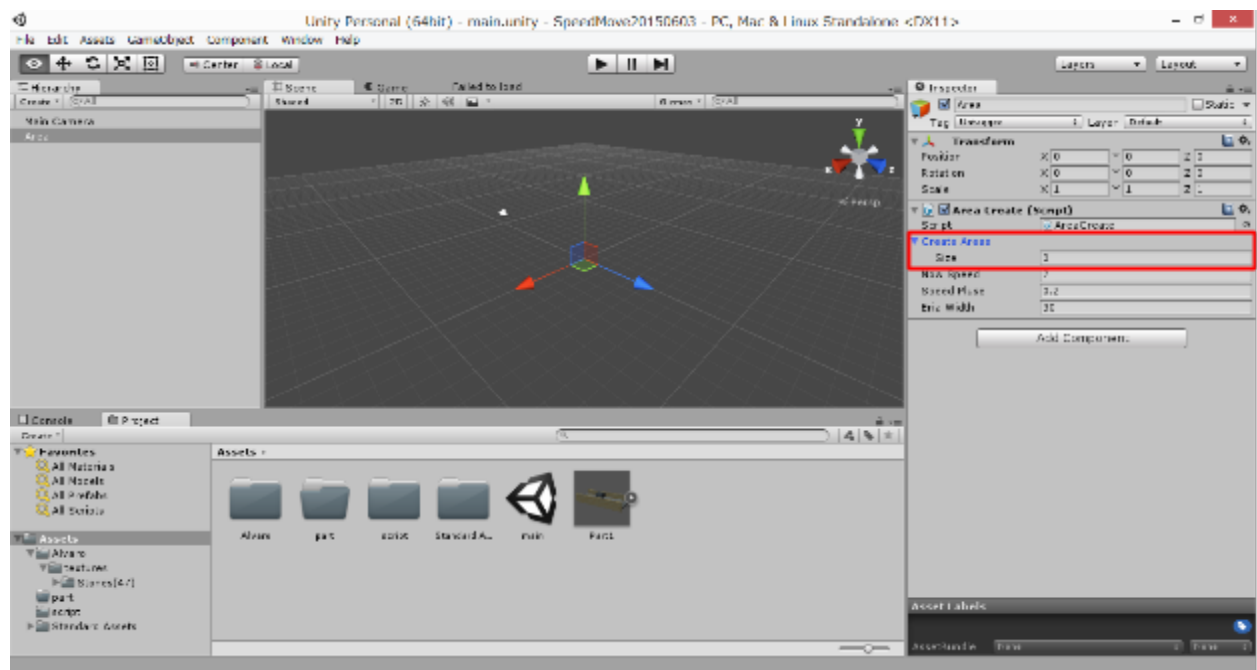


その後Projectウィンドウの Assets>part までドラッグしてプレファブ化
HierarchyビューのPart1は削除してください。



3-3) プレファブ化したオブジェクトを設定
AreaのAreaCreateのCreateAreasのSizeを 1 にElementにProjectのPart1 をド

ラッグ
ゲームを起動すると設定したPart1が自動生成される。



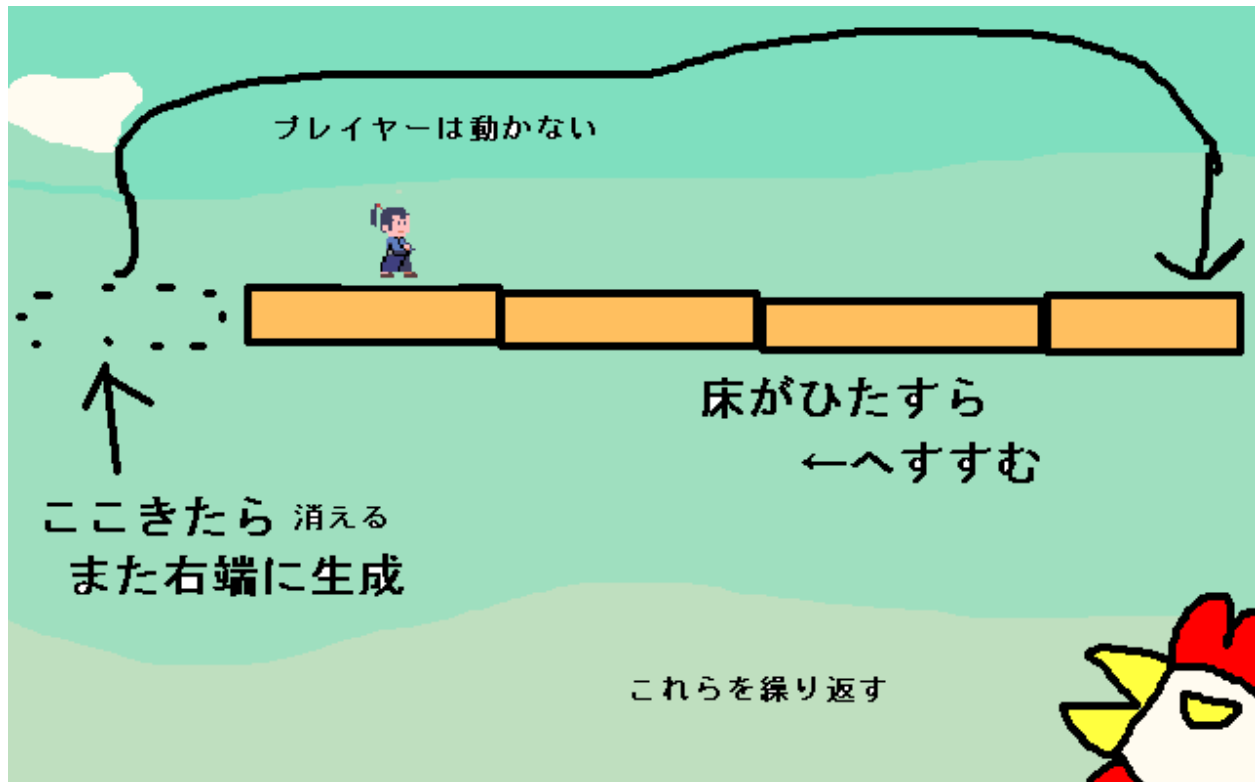
※自動生成の仕組み

10個のエリアをCreateAreasのなかからランダムで選出し配置します。

全体をキャラクター（カメラ）に向かって手前に動くようにすることで自分が移動しているように見せます。

ある程度手前へ移動したエリアは消去され奥に出現します。

それをループすることで無限にエリアを移動していくように見せます。



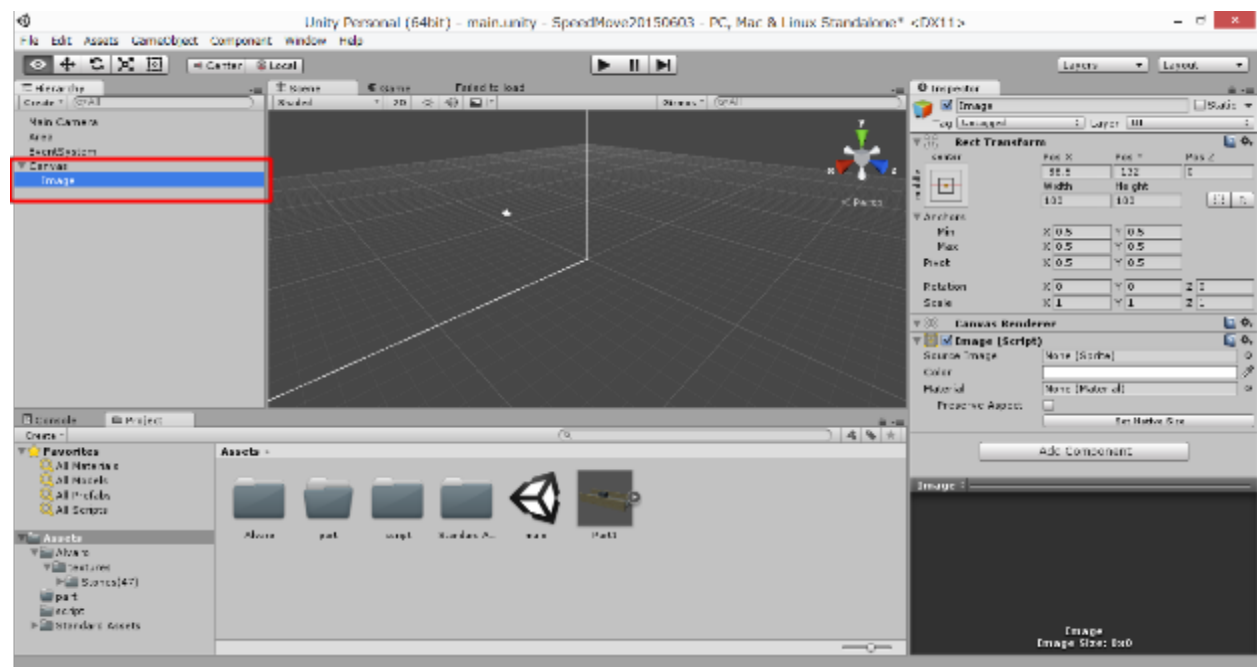
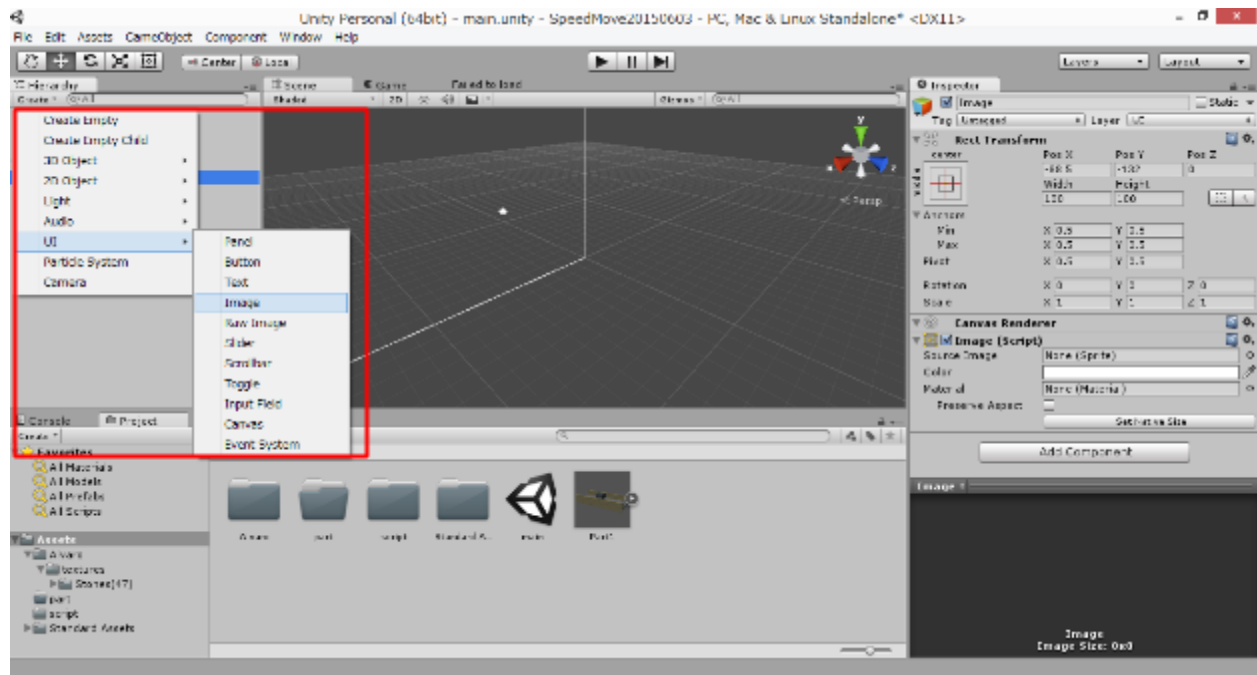
●4点数の表示

4-1)文字表示の枠と文字を作成

Hierarchy>Create>UI>Image で枠を作成。

Hierarchy に

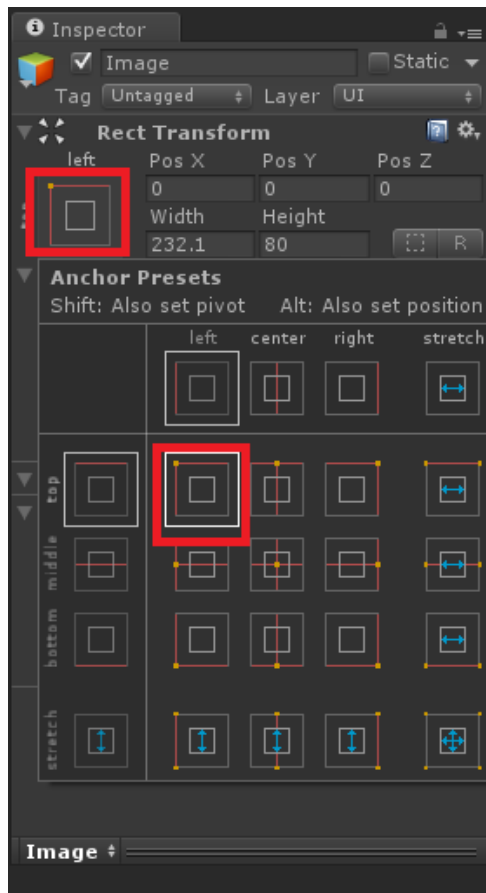
Canvas> Image という親子オブジェクトが出来きる。



4-2)左上寄に変更

ImageをクリックしてInspectorを開く。

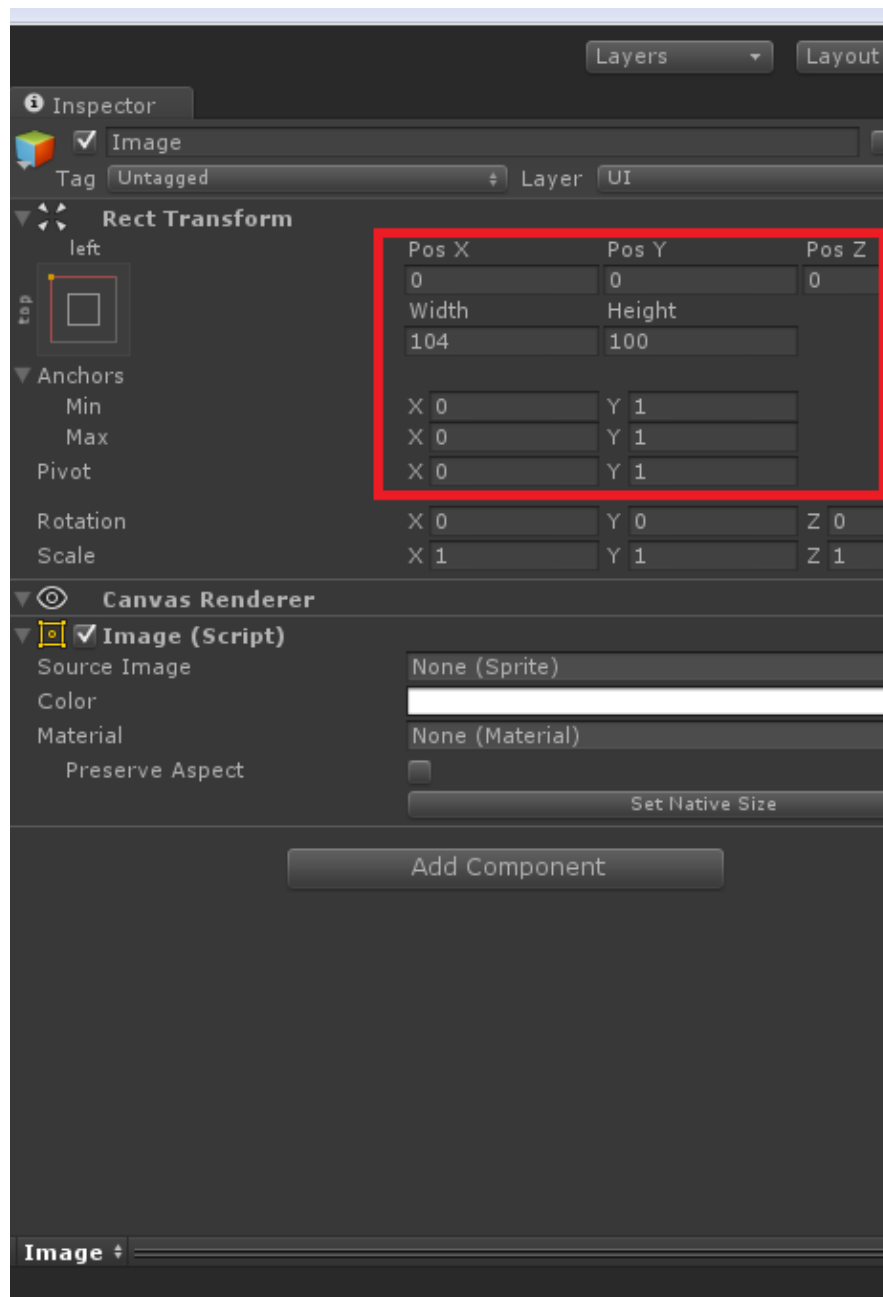
RectTransformの項目の
左のほうに四角があるのでクリック
その後 **Top Left** を選択



4-3) Anchors や 位置を設定
Anchors左の▼を押して展開
Min X:0 Y:1
Max X:0 Y:1
Pivot X:0 Y:1

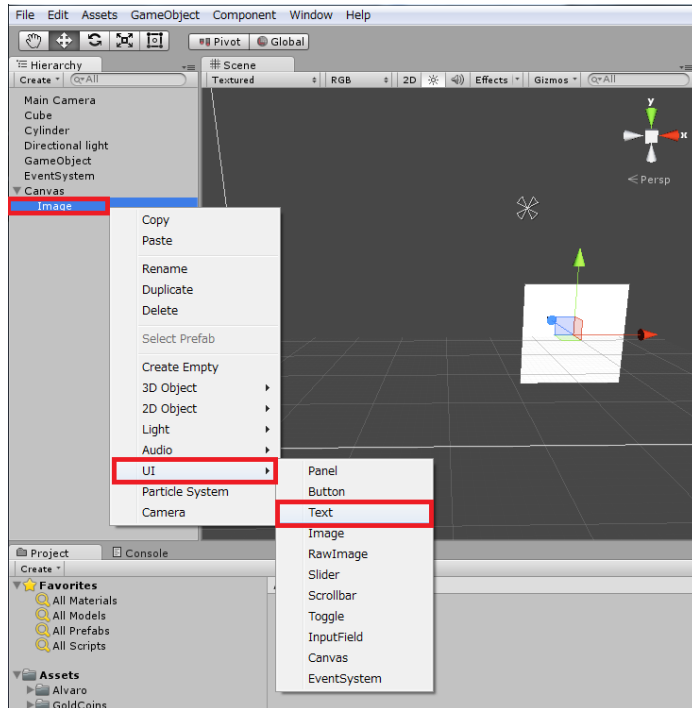
PosXを0 PosY を0に設定

※必ずAnchorsを設定した後にPosを設定！



4-4) Text（点数）の作成

Hierarchy>Create>UI>Text でText（文字）を作成



- ・ 名前はScoreに設定

RectTransformの項目の

左のほうに四角があるのでクリック

その後 Top Left を選択

Anchors項目内

Min X:0 Y:1

Max X:0 Y:1

Pivot X:0 Y:1

PosXを0 PosY を0に設定

4-5)Speed（速度）の作成

Hierarchy>Create>UI>Text でText（文字）を作成

- ・ 名前はSpeedに設定

RectTransformの項目の

左のほうに四角があるのでクリック
その後 **Top Left** を選択

Anchors項目内

Min X:0 Y:1

Max X:0 Y:1

Pivot X:0 Y:1

PosXを0 PosY を-27に設定

4-6) ついでにGameOver作成

Hierarchy->Create->UI_>Text でText（文字）を作成

- ・ 名前はGameOverに設定

PosXを0 PosY を0に設定

Textを GameOverに変更

4-7) 点数表示のスク립トを導入

Canvasをクリック Inspectorが開く

一番下にあるAddComponent

Scripts>score

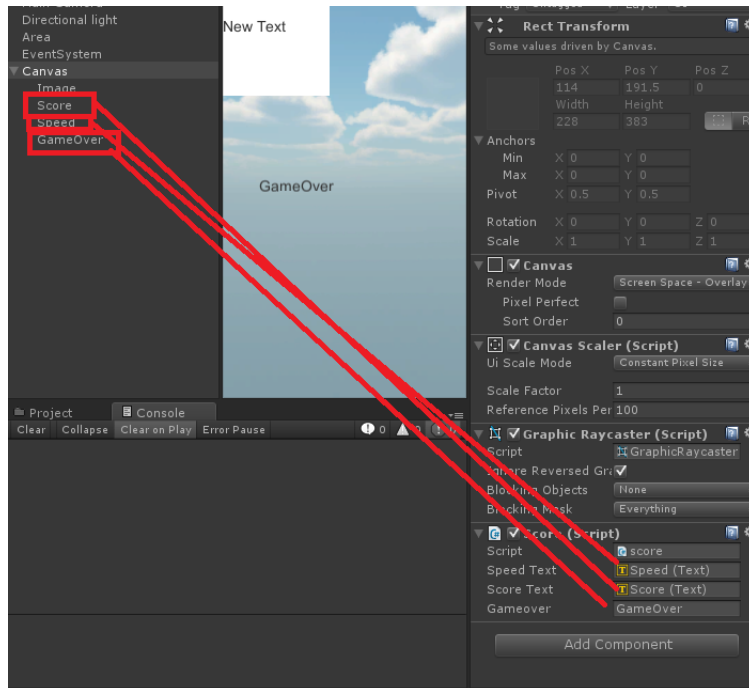
Speed Text に対してさっき作ったSpeedをドラッグして設定

Score Text に対してさっき作ったScoreをドラッグして設定

GameOver に対してさっき作ったGameOverをドラッグして設定

点数によって各Textが更新されるようになる。

GameOverになると GameOverが表示される



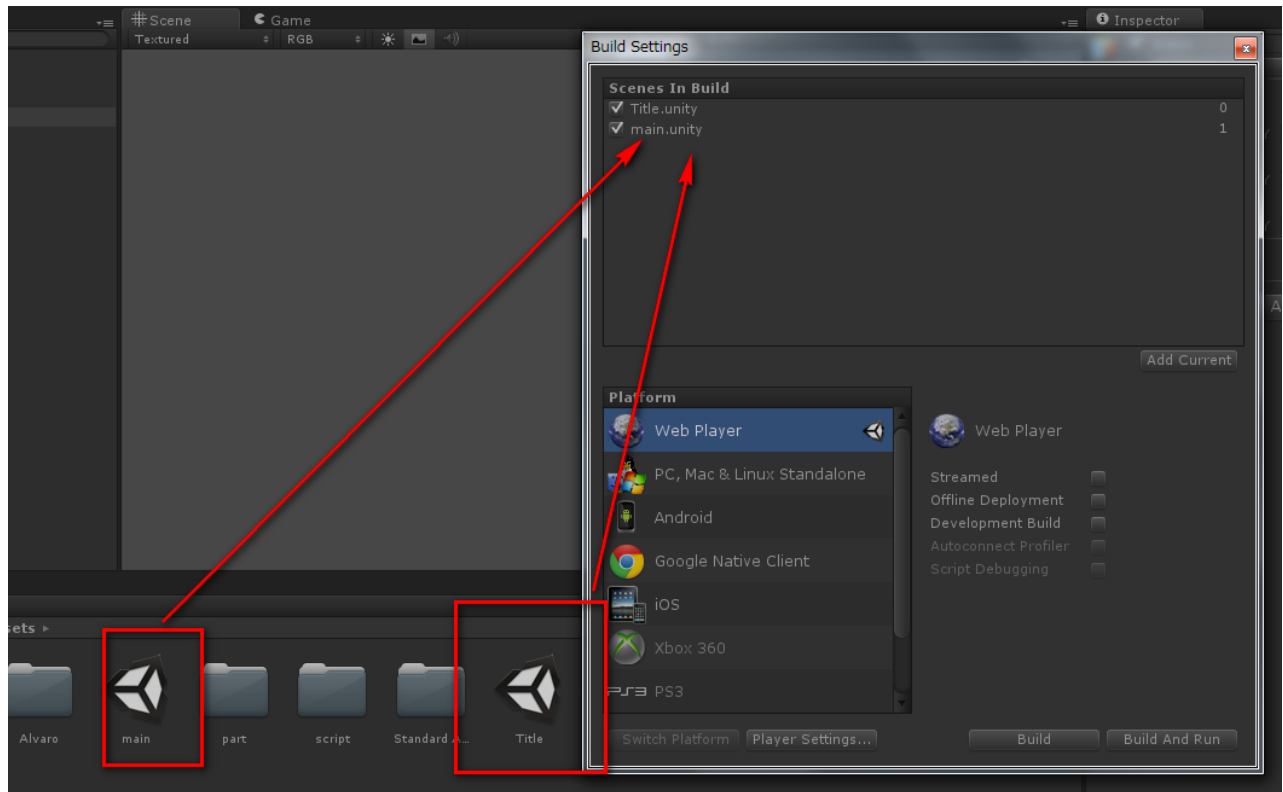
●5タイトルの表示

5-1)シーンを作成

上のメニュー File>NewScene

新しいシーンを作る なまえはTitle

File->BuiledSetting



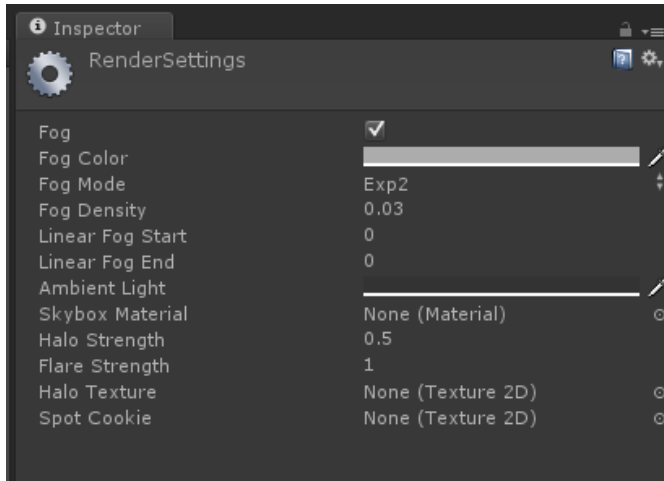
5-2)シーンの登録をします。

MainCameraにTitle.csを設定
AddComponent>Scripts>Title

5-3)文字を作成

Hierarchy>Create>UI>Text でText（文字）を作成
PosXを0 PosYを0に設定
textにPushStartと入力

6)奥行き感をだす霧をだす
[Window]->[Lighting]
Fog をONに
FogDensityを0.03をに設定
FogColorはお好みで



そのほか追加してみると面白そうなもの

- ・ 四角以外のトラップ

アセットストアからいろんな3D素材を落としてエリア上に配置してみる。

- ・ 動くトラップ

おいているオブジェクトにあたると終了というだけでは面白くないので一定の動きをするオブジェクトを用意してみよう。

- ・ 落下トラップ

床に穴をあけてそこに入ったら落ちて死亡

MainCameraに重力を入れて一定以下まで座標が下がったら死亡判定とか

- ・ ゲームオーバー演出

画面を薄い赤のフィルターをかける。

画面を揺らす。

- ・ 横はぶつかってもオッケー

正面からぶつかったらゲームオーバーだが、横の場合ははじかれるだけという処理を作ってみよう。

実装例

横用のあたり判定を用意して、それに壁が触れたらはじかれるとか。

- ・ ジャンプ

カメラをジャンプさせる。

GUIボタンを押したらジャンプ・またはダブルクリックなどでジャンプ

- ・ 砲弾を打って障害物を蹴散らす

GUIボタンを用意する？弾数を制限？