3. 開発環境調査

3.1 開発環境を知る

　世の中には数多くのプログラミング言語が作り出されています。一説にはその数、何万種類を超えるともいわれています。なぜそんなにも沢山の種類の言語が必要なのでしょうか。

　その理由はいくつも考えられますが、大きなものとしては「モダンな言語に進化してきた（新しい機能や考え方が取り入れられて歴史的に精錬されてきた）」ことや「開発したいアプリの特性に沿った言語が求められている（Webページ用アプリと3Dゲーム機用アプリでは使いたい命令が大きく異なる）」などがあるかと思います。

　また、使いたいハードウェア等の環境（プラットフォーム）や使いたい言語、作りたいアプリケーションの種類、開発の規模などによって様々な開発環境が提供されています。Windowsアプリを作るためのVisual Studioや、Java言語でアプリケーションを作るために使われるEclipseやIntelliJ IDEAなどのいわゆる統合開発環境なども開発環境のひとつです。そして作る対象に合わせて、開発環境に様々なツール（SDKなど）の追加も必要になってきます。

　皆さんが自分でアプリケーションを作る上で、以上のようなことをしっかりと考える必要があります。つまり自分が作りたいアプリの

・動作プラットフォームは？（Web、スマートフォン、コンシューマゲーム機、etc）

・アプリの種類は？（ゲーム、エンターテイメント、実用系、システムツール、etc）

・使いたい機能は？（2Dグラフィックス、3Dグラフィックス、センサー、DB、etc）

・開発言語は？（Java、JavaScript、C#、Ruby、Python、Lisp、etc）

などを考えなければいけません。そしてそれらに合わせた開発環境を自分で構築し、使いこなしていく必要があるわけです。

　開発環境も様々なものがありますが、いくつか調査して自分にあったものを見つけ出すようにしましょう。この授業の中でもいくつかの開発環境に触れてみます。皆さんも、他にも自分で色々と使ってみたい開発環境や言語を探してみてください。

3.2 Unity

　Unityは「Unity Technologies」によって開発・販売されているゲーム開発環境です。

Unity TechnologiesのWebサイトから抜粋（<http://japan.unity3d.com/unity/>）2013.1.9

|  |
| --- |
| **Unityで夢のゲーム作りを**  　Unity はゲーム開発のエコシステム。インタラクティブな 3D コンテンツ作成のための直感的なツールとサクサク作れるワークフローが、パワフルなレンダリングエンジンと完全に統合された環境で開発が出来ます。  　手軽なマルチプラットフォームパブリッシング、高品質なアセットが幾千も揃ったアセットストア、困ったときにノウハウを共有してくれる巨大なコミュニティがあなたを支えます。  　Unityの民主的なエコシステムは、美しく手の込んだゲームを作り出す時に、多くのデベロッパーやスタジオが直面する障壁を突き崩す手段となるでしょう。彼らは、人々が夢中になって喜ぶゲームをあらゆるプラットフォームで作るために、Unity を活用しているのです。 |

　現在、様々な有名ゲームがUnityを利用して作られて販売されています。

　Unityはマルチプラットフォーム対応が進んでおり、PC用のソフトウェアから各種スマートフォン（iPhone(iOS)、Android）、各種ゲーム機（PS3、Wii U、XBOX360など）などで動くアプリケーションがUnityで開発されています。

　Unityで一番特徴的なこととして『物理演算のシミュレートを手軽に行える』があります。通常の開発環境（Eclipse＋java等）を使って「ボールが落下して地面に跳ね返る」という画面をプログラム言語で描画させるには、時間遷移による「ボールの位置の計算」「画面の計算と描画」「ボールと床の当たり判定」「跳ね返り係数等に従っての新しい移動方向と量の計算」など様々な計算と描画を行うアルゴリズムをプログラマが考えて実装しなければなりません。しかしUnityを使うとこれらの計算（＝物理演算）はUnityが行ってくれるので、プログラマはそれら以外のアルゴリズムを考えることに集中ができます。

　Unityには様々なバージョンがあります。無料で使えるバージョン（UnityとかUnity Freeと呼ばれています）もありますので、アプリケーション開発の勉強としてUnity開発環境に挑戦をしてみましょう。

|  |
| --- |
| **【演習：Unityのセットアップと構成の確認】**  Unity開発環境のセットアップを行い、サンプルプログラムを確認してみましょう。  1. Unityの入手とインストール  (1)Unity Technologies Japanのサイトにアクセス  <http://japan.unity3d.com/>  (2)「Download」のページで「Windows用無料フルバージョン」のUnityをダウンロード  (3)ダウンロードしたファイルを実行してFree Versionでインストール  　　Pro(有料版)やTrial（試用版）などと間違えないようにインストールしてください。  　　Unityのユーザ登録していない場合は「Crate Account」する必要があります。  2. 付属のサンプルプロジェクトを確認する  デモンストレーション用のゲームが用意されていますので、それを確認しましょう。  (1)「Open Project」→「Open Other…」をクリック  (2)「ドキュメント」フォルダにある「Unity Projects」フォルダを開く  (3)「4-0\_AngryBots」のフォルダを選択して「フォルダーの選択」ボタンをクリック  (4)読み込みが終了したらウインドウ上部にあるPlayボタン（右向き三角形）を押す  ※カーソルキーでプレイヤーキャラの移動、マウス操作で弾の発射  Playボタンを再度押すとゲーム終了  3. 開発画面の確認  Unityの画面は主に次のようなビュー（サブウインドウ）から構成されています  　　○Hierarchyビュー（ヒエラルキービュー）  　　　　オブジェクト（物体）がどのようなオブジェクトから構成されているかの  　　　　階層構造を確認できます。  　　○Sceneビュー（シーンビュー）  　　　　3D空間が表現されていて、オブジェクトの配置などを設定できます  　　○Gameビュー（ゲームビュー）  　　　　ゲーム画面が表示されるビューです  　　○Projectビュー（プロジェクトビュー）  　　　　プロジェクトを構成するファイルが一覧表示されています  　　○Consoleビュー（コンソールビュー）  　　　　開発環境からのメッセージ等が表示されます  　　○Inspectorビュー（インスペクタービュー）  　　　　オブジェクトやファイルに対する様々な設定を変更する事ができます    4. サンプルプロジェクトをもとにオブジェクトの設定を確認する  敵やプレイヤーなどのオブジェクトがどのように構成されているかを見てみましょう  (1)Hierarchyビューで「Enemies」を開く  (2)開いて出てきた「EnemySpider」などをダブルクリック  (3)更に開いて、そのオブジェクトがどのようなオブジェクトで構成されているかを確認  　（たとえば「AI」というオブジェクトには人工知能となるプログラムが記述されています）  ※他のオブジェクトについても色々確認してみてください  (4)Projectビューで「Texture」をクリック  　（使われているテクスチャーが表示されます。色々見てみてください）  (5)どれかテクスチャーを右クリックして「Open」を選ぶ　（画像を確認できます）  (6)Projectビューで「Scripts」→「AI」と開く  (7)JSと書かれたアイコンどれかをクリック　（JavaScriptで書かれたコードが見られる） |

|  |  |
| --- | --- |
| **【演習：初めての開発（Unity）】**  Unity開発環境でアプリケーション開発を試行してみましょう。  1. プロジェクトの新規作成と開発環境のレイアウト設定  (1)「File」メニューから「New Project...」を選ぶ  (2)「Create」ボタンをクリック  (3)「Window」メニューから「Layout」→「Wide」を選ぶ  ※「Wide」に限らず自分で使いやすいLayoutを選んでもいいでしょう  2. オブジェクトの作成  (1)Hierarchyビューから「Create」→「Cube」を選ぶ　（Cubeは直方体のことです）  (2) 上のほうにあるボタン群から「スケールツール」を選ぶ  視点移動ツール  平行移動ツール    スケールツール  回転ツール  (3)赤・緑・青の四角形をドラッグして、以下のような薄い板状に形を変える    (4)ボタン群から「平行移動ツール」を選ぶ  (5)赤・緑・青の円錐をドラッグして、板を移動させる  　（Playボタンを押したときにGameビューで板がすべて表示されるように移動させる事）  (6)Hierarchyビューから「Create」→「Sphere」を選ぶ　（Sphereは球のことです）  (7)作られたSphereを選んだ状態でInspectorビューのScaleのX,Y,Zに全て5を入れる  　（スケールツールで行っても良いですが綺麗な球にするには値入力のほうが簡単です）  (8)ボタン群から「平行移動ツール」を選ぶ  (9)赤・緑・青の円錐をドラッグして、板を移動させる  　（Playボタンを押したときにGameビューで板がすべて表示されるように移動させる事）  (10)Sceneビューの右上側にある直方体と円錐の組み合わさったものをクリックすると、様々な方向からオブジェクトを確認することができます。板の真上に球があるように球の位置を移動させましょう    (11)Hierarchyビューから「Create」→「Direction light」を選ぶ  　（Direction lightは平行光源のことです）  (12)PlayボタンをクリックしてGameビューで表示されるものを確認しながら、板と球が明るく表示されるようにDirection lightを移動させる  3. オブジェクトに物理法則を定義する  物理法則に従ってオブジェクトが移動したり衝突して反射したりするようにします  (1)Hierarchyビューに表示されているSphereを選ぶ　（球が選択状態になります）  (2)Componentメニューから「Physics」→「Rigidbody」を選ぶ  　（InepectorビューにRigidbodyという項目が追加されます）  　（Physicsは物理、Rigid bodyは剛体という意味です）  (3)Playボタンをクリックして動作確認してみましょう  　（球が落下して板の上に落ちて止まるはずです）  (4)HierarchyビューからCubeを選んで、Componentメニュー→「Physics」→「Rigidbody」  (5)InspectorビューのRigidbodyという部分にある「Is Kinetic」にチェックを入れる  　（「Is Kinetics」にチェックを入れることで板は落下しないけど動けるようになります）  (6)Projectビューから「Create」→「Physic Material」を選ぶ  　（ProjectビューのAssets欄に「New Physic Material」というアイコンができます）  (7)「New Physic Material」を選び、Inspectorビューの「Bounciness」の値を1にする  　（反発係数(Bounciness)が100%の物理定義(Physic Material)を作成しました）  (8)Hierarchyビューに一覧表示されているCubeを選ぶ  (9)「New Physic Material」のアイコンを、Inspectorビューの「Box Collider」という文字の欄にドラッグ＆ドロップする  (10)Hierarchyビューに一覧表示されているSphereを選ぶ  (11)「New Physic Material」のアイコンを、Inspectorビューの「Sphere Collider」という文字の欄にドラッグ＆ドロップする  　（これで先ほどの反射係数に関する物理法則が板と球と適用されました）  ※ Playボタンを押して動作確認してください。球が板に当たって跳ね返ればOKです  4, 板をカーソルで操作可能にする  JavaScript言語でプログラムコードを記述します  (1)Projectビューから「Create」→「Javascript」を選ぶ  　（ProjectビューのAssets欄に「NewBehaviour」というアイコンが作成されます）  (2)「NewBehaviour」アイコンをダブルクリック  　（プログラムを記述するエディターが起動します）  (3)プログラム中のUpdate()の部分を以下のように記述して保存する   |  | | --- | | function Update () {  transform.position.x +=  Input.GetAxis("Horizontal") \* (50.0 \* Time.deltaTime);  transform.position.z +=  Input.GetAxis("Vertical") \* (50.0 \* Time.deltaTime);  } |   (4)「NewBehaviour」のアイコンをHierarchyビューのCubeにドラッグ＆ドロップする  ※ Playボタンを押して動作確認してください。板が移動すればOKです  　【プログラムの解説】  　　　transform.position.x　　…　X座標位置  　　　transform.position.z　　…　Z座標位置  　　　Input.GetAxis("Horizontal") …　左右カーソルキーの入力  　　　Input.GetAxis("Vertical ") …　上下カーソルキーの入力  　　　Time.deltaTime　　…　この関数が呼び出される感覚の時間（秒）  5.保存  (1)Fileメニューから「Save project」を選ぶ  6. ビルドして実行する  (1)Fileメニューから「Save Scene」を選ぶ  (2)「BallGame」と入力して「保存」する  (3)Fileメニューから「Build & Run」を選ぶ  (4)「Add Current」ボタンをクリック  (5)「Build & Run」ボタンをクリック  (6) 「BallGame」と入力して「保存」する  (7)ディスプレイ解像度を選んで「Play!」ボタンをクリック  ※(6)で作られたexeファイルを実行すればいつでもPlayできます |

応用練習

　・板の大きさを調整して、ゲームバランスを最適化してみましょう

　・板の移動スピードを調整してみましょう

　・複数のボールを落下させて跳ね返すようにしてみましょう

　・板を上下（座標軸でいうところのY方向）にも移動できるように変更しましょう

　・板の反発係数を調整してみましょう

　・反発係数の異なる複数の板を同時に移動させてみましょう

　・その他、Unityのメニューを色々と触ってみましょう

　Unityで作られたアプリケーションが色々と公開されています。WebサイトやGitHubで探してみましょう。他にも、Unityの開発に書かれているブログなどを探して色々と調べてみてください。