

Unity班 活動報告書

立命館コンピュータクラブ
2020 年度プロジェクト活動

2021 年 2 月 8 日

堀越 俊行^{*1}新藤 尚輝^{*2}宇佐 基史^{*3}
原 佑馬^{*4}北村 優奈^{*5}中山 凌一^{*6}
服部 瑠斗^{*7}岡本 陽太^{*8}桐井 優実^{*9}
中川 拓海^{*10}伊藤 佑^{*11}山本 京介^{*12}

^{*1}情報理工学部 情報理工学科 システムアーキテクトコース 三回生

^{*2}情報理工学部 情報理工学科 実世界情報コース 二回生

^{*3}理工学部 ロボティクス学科 二回生

^{*4}情報理工学部 情報理工学科 システムアーキテクトコース 三回生

^{*5}情報理工学部 情報理工学科 システムアーキテクトコース 三回生

^{*6}情報理工学部 情報理工学科 システムアーキテクトコース 三回生

^{*7}情報理工学部 情報理工学科 知能情報コース 三回生

^{*8}情報理工学部 情報理工学科 実世界情報コース 三回生

^{*9}情報理工学部 情報理工学科 システムアーキテクトコース 二回生

^{*10}情報理工学部 情報理工学科 セキュリティ・ネットワークコース 三回生

^{*11}理工学部 三回生

^{*12}情報理工学部 情報理工学科 システムアーキテクトコース 一回生

目 次

1	初めに	3
1.1	プロジェクト概要	3
1.2	Unity とは	3
2	活動について	3
3	個人の制作物紹介	4
3.1	Test Game	4
3.2	アクションゲームにおける攻撃の実験環境	6
3.3	EscapeMaze	8
3.4	StudyTyping	10
3.5	Mysterious Fortification	12
4	活動の上での問題点	14
5	活動で得られた知見	15
6	班総括	15

1 初めに

文責: 宇佐 基史

1.1 プロジェクト概要

当プロジェクトは、Unity の知識を深め、実際に制作を行う事でプロダクト作成に必要な知識を獲得することにある。具体的な活動内容は後述とする。

1.2 Unity とは

Unity Technologies 社が開発しているゲームエンジンである。特徴として、100 万を超える利用者を有する点、一定の条件下において無料で使用できる点、多くのプラットフォームに対応可能である点などが挙げられる。加えて、物理演算や VR などにも利用可能である。

2 活動について

文責: 北村 優奈

活動について、春セメスターは週 2 回集まり Unity の基礎の学習と企画の発表、ゲームの制作を行った。秋セメスターは週 1 回集まり各自プロダクト作成の進捗の確認と次週までの目標の設定を行った。企画では XR アプリケーションの作成を希望する場合、秋セメスターから XR のプロダクトを作成を開始する予定であったが、希望者が居なかったため 1 年を通して Unity の勉強とプロダクト作成を行う形になった。以下に春セメスターで行った活動内容のうち代表的な部分の詳細を報告する。

2.0.1 Physics

Physics とは、Unity で物理演算を動かすための機能である。空間を探索して衝突などを検知することと、オブジェクトを実際に動かすことという二つが主な役割である。オブジェクトに機能を実装するコンポーネントとして、当たり判定検出用の Collider と、重力など物理特性によって制御出来るようにするための Rigidbody などがある。

2.0.2 Cinemachine

Cinemachine とは、Unity のカメラを制御するためのコンポーネントである。Unity2018 から追加された PackageManager を使うことでプロジェクトに Import して使用することが出来る。使用することで、ターゲットの動きに合わせたカメラの制御や、複数のカメラによるカットの切り替え、レールを敷いたように水平に撮影するドリー撮影などをすることが可能になる。

2.0.3 UnityEventFunction

UnityEventFunction とは、Unity がアプリケーションを実行する際に扱う予約済み関数のことである。具体的には最初のフレームを更新する前に呼び出される Start() や、1 フレームにつき 1 度呼び出される Update(), オブジェクトの衝突の際に呼び出される OnCollision() などがある。

3 個人の制作物紹介

3.1 Test Game

文責: 新藤 尚輝

3.1.1 概要

個人制作物として、3DCG のアクションゲームを作った。このゲームを作ったきっかけは、PS4 のコントローラーを使ってキャラクターを操作したかったからである。このゲームには基礎的な部分に加えて、細かい所に工夫がしてある。

3.1.2 作品の紹介

このゲームは、主に以下のオブジェクトから作成されている。

- DreamForestTree(地形)
- unitychan(プレイヤー)
- Zombie Cartoon(敵キャラクター)

上記の三つは共に unity の「Asset Store」から拝借したものである。また、このゲームには一部効果音・BGMが入っている。それらについて、以下に箇条書きにする。BGM は「魔王魂」、SE は「無料効果音で遊ぼう！効果音検索」のものを利用した。

- ファンタジー 15(BGM)
- ファンタジー 13(BGM)
- short_punch1(SE)
- kick1(SE)

このゲームは、以下のボタンを用いて操作する。

- ジャンプ (○ボタン)
- 画面遷移 (Options ボタン)
- 攻撃 (R1 ボタン)
- スライディング (×ボタン)
- ロックオン (R3 押し込み)
- カメラ操作 (R スティック)
- 移動 (L スティック)

図 1 は、このゲームのタイトル画面である。この画面で、Options ボタンを押すと、ゲームが開始する。

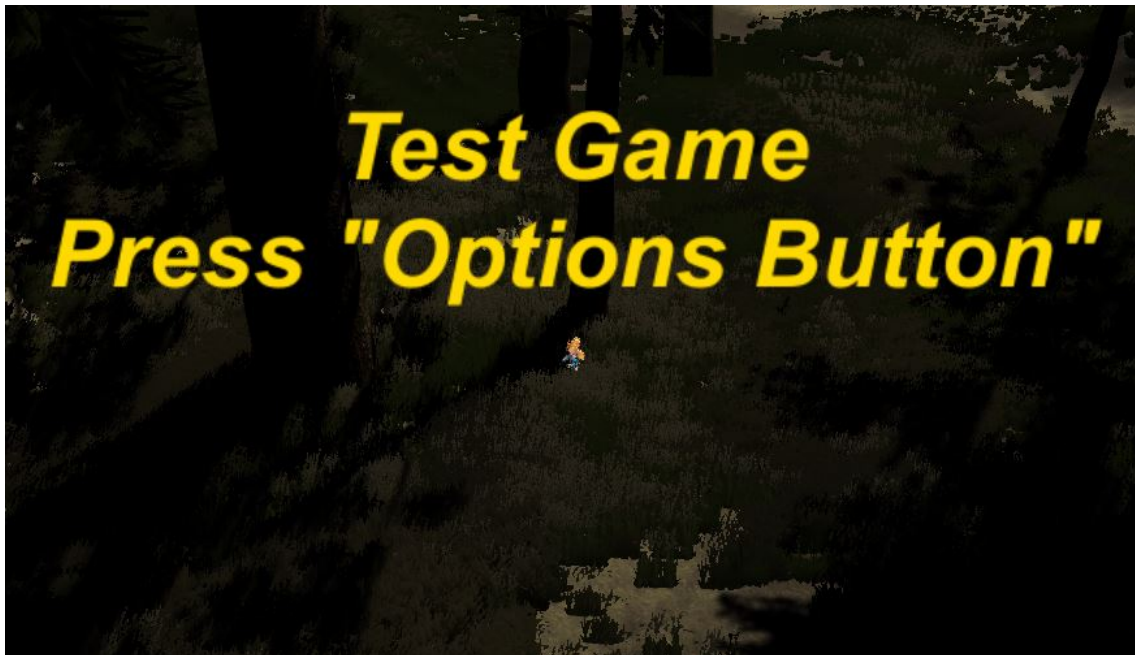


図 1: タイトル画面

このゲームは敵キャラクターを 10 体倒すとゲームクリアとなる．図 2 の画像が，クリア画面である．



図 2: クリア画面

また，操作キャラクターの HP が 0 になると，ゲームオーバーとなる．図 3 の画像が，ゲームオーバーの画面である．この状態で PS4 の options ボタン押すと，図 1 のタイトル画面に戻る．



図 3: ゲームオーバー画面

以上のような流れで、このゲームが進行する。

3.1.3 展望

今回作成したゲームは、コード以外の BGM やモデル、モーションは全て他者の著作物である。この様に借りた物で作るのではなく、自身で作成したモデルやモーションと今回作成した基礎を基に、独自のゲームを作成したい。

3.1.4 感想

ロックオン機能を設けたののだが、作成途中に多くのバグが発生して、大変だった。また、カメラが荒ぶることも多く、非常に悩ましかった。これらの機能については、未だに改善中だが、Unity に関する知識は、かなり深まったと感じている。

3.2 アクションゲームにおける攻撃の実験環境

文責: 宇佐 基史

3.2.1 概要

アクションゲームにおける攻撃の基礎単位を見つめなおす実験環境の製作を目指した。報告書には現状として、Unity の 2D ゲーム制作環境を学ぶ過程で制作されたシューティングゲームと、後述する移動機能を除くパラメータ調整をするシステム部分を記載することとする。

3.2.2 作品の紹介

「概要」にある事から、制作前の企画書と現状のプロダクト概要を並行した内容とする。

企画書概要 ゲームにおける攻撃の要素を以下と定義する。

- 入力から攻撃の発生までの時間
- 攻撃の発生から再入力が可能になるまでの時間
- 攻撃前後の硬直時間
- 攻撃の範囲
- 攻撃の威力

これらの要素にキャラクターの移動性能とサイズのパラメータを加えた物をゲームの基礎要素と考える。全ての要素の中で威力を除いたパラメータをゲーム開始前のメニュー画面でプレイヤーが各自設定できるようにする。パラメータの設定にはコストを設け、有利な値の設定にはコストを高く、逆に不利なパラメータ設定にはコストを低く付加するものとする。コストをキャラクターそのものの強さを表す指標となるようにする。二人プレイの対戦型ゲームである。一キャラクターあたり、二種の攻撃を設定できるようにする。2人で相打ちの場合は攻撃の無効化と攻撃の逆方向への移動が付加される。HP(ヒットポイント)の概念は無く、一度の攻撃を相手に当てることで1ポイント獲得とし、3点先取をゲームの勝利条件とする。一人につき以下の操作系を実装する。

- ボタン 1(攻撃 1or 決定ボタン)
- ボタン 2(攻撃 2or 決キャンセルボタン)
- 移動の 8 方向入力

音声・画像などのアセットを極力自分で用意する。

プロダクト概要

シューティングゲーム 敵、当たり判定、それに伴う演出、キー入力の獲得、マウス座標の獲得といったシステムが実装されている。なお、制作におけるアセットは全てフリー素材を用いている。完成の際には次の項目で用いる独自の物を実装する。既出の Web サイトを参考に Unity の理解を目的の制作である。

パラメータ調整 攻撃範囲は扇形、半径と中心角の角度で決められる。自機の慣性力、加速力、最高速度も同時に設定する。制作におけるアセットは全て自作する。シンプルを基調に考える。

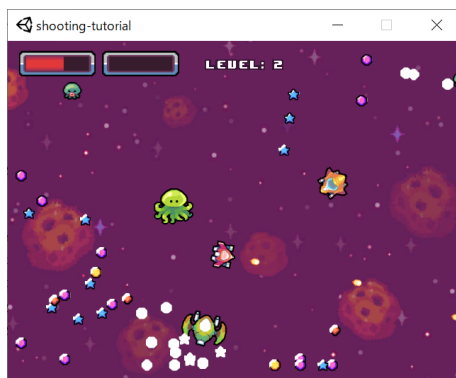


図 4: シューティングゲームシーンその 1

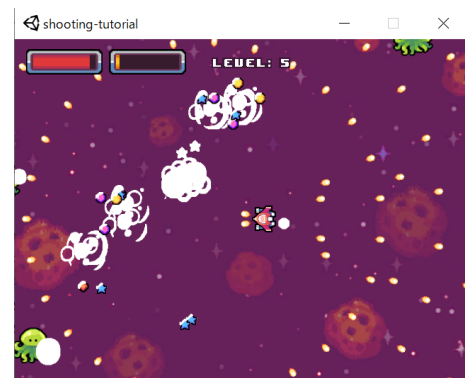


図 5: シューティングゲームシーンその 2

3.2.3 展望

技術面 まずは完成にこぎつけること。PC のスペック不足のみを原因と解釈しないようにする。メモリ管理やバージョンの違いによるエラーに対応できるようにする。ゲームはレスポンスの理解としての、CPU 付加に対する知識の獲得。CPU の作成 (≒一人用モード)。

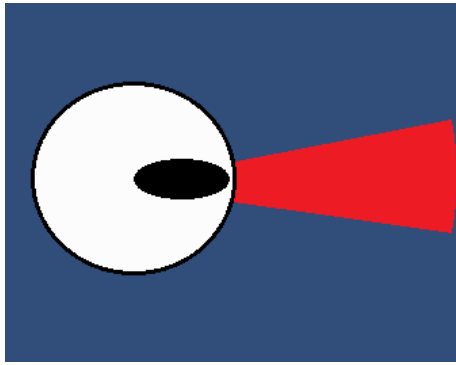


図 6: システム部分 (攻撃範囲：狭)

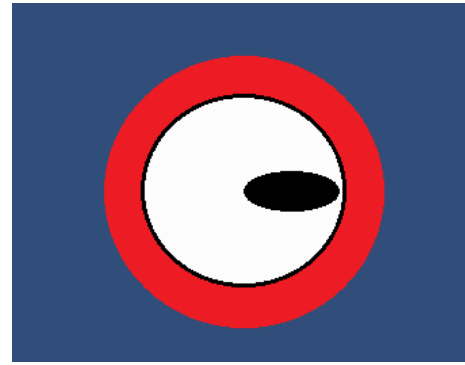


図 7: システム部分 (攻撃範囲：広い)

企画面 CPU の作成 (≡一人用モード)。世間一般に、重量級キャラ、遠距離キャラ、近接キャラなどの俗称を持つキャラクターいる。それらすみ分けがパラメータのどの部分に起因するのか、それが理解できるようなゲームシステムの実現を目指す。最終的には勝ち抜き型の一人用モードを実装し、コストやパラメータ設定に規制 (≡縛り) を設ける。その縛りの中でプレイヤーに勝つ方法を模索してもらおうといったゲームを作り上げたい。

3.2.4 感想

PC の不調により当初予定したものが作れなかったのが悔やまれる。個人の見解としては現行の 3D ゲームは多くの場合、重力に縛られたキャラクターが用いられている、宇宙空間を題材にした物でも上下の概念がある、ジャンプのような非平面的な移動も平面的なアクションのバリエーションとして扱われている、といったものである。つまり、3D ゲームと分類される物であっても、「(カメラワークの問題を除いて) 基本的なシステムは 2D ゲームを階層的積み上げた物にすぎない」と解釈できる。このゲームを触る事であらゆるアクションゲームの駆け引きの核を捉えることができるようになれると構想していた。人間も地図のように三次元空間を二次元として解釈し、移動を平面的と捉える側面はある。その中で本当の意味での 3D ゲームとはどういった物になるのか、遠い未来に形にしてみたい。

3.2.5 主に参考にしたサイト

【Unity 入門】60 分で作るシューティングゲーム (最終閲覧日：2021 年 2 月 6 日)

<https://nn-hokuson.hatenablog.com/entry/2016/07/04/213231>

【Unity 2020 入門】全方位シューティングを作る 全 26 回 (最終閲覧日：2021 年 2 月 6 日)

<https://baba-s.hatenablog.com/entry/2018/04/01/190000>

3.3 EscapeMaze

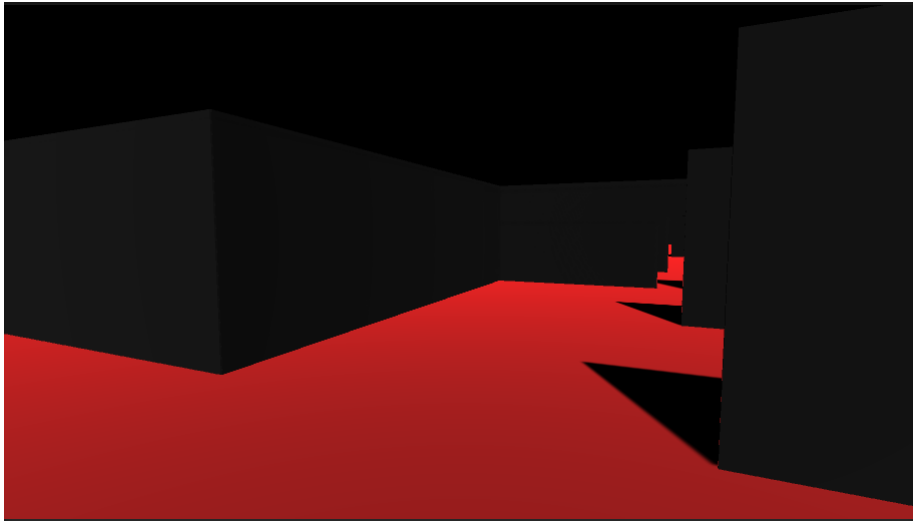
文責: 原 佑馬, 山本 京介

3.3.1 概要

迫りくる敵から逃げながら迷路内で謎解きをしながらアイテムを集め、脱出を目指す 3D ホラーゲームである。

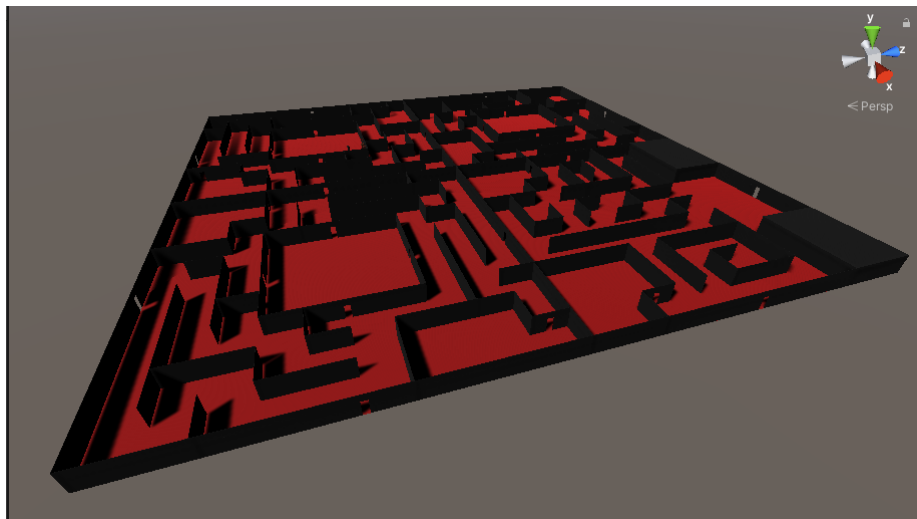
3.3.2 作品の紹介

一先ず作品として形にするため、プレイヤーがマップ内を移動できるようにした。



プレイヤーの移動はキーボード (WASD), 視点の移動はマウスで行う。

マップは以下のような構造である。なおこのマップは Blender というモデリングソフトで簡単に作成した。



アイテムやそれを配置する棚もモデリングした。本来はこれらのアイテムを回収しつつ、脱出を目指す。



マップの探索中は以下のような敵が追いかけてくる。



3.3.3 展望

まだ初歩的な部分しか実装されていない。アイテム回収や謎解きシステム、そしてゲームのクリア時の挙動などを実装し本格的なゲームに近づけたい。

3.3.4 感想

Unity での共同制作は初めてだったが、Slack で意思疎通を図ったり、GitHub で制作物の管理をしたりすることで共同作業で発生する齟齬はできるだけ抑え込めたのではないかなと思う。ただゲーム制作のスキルの差から一任する作業も多く、結果的に進捗が芳しくなかったことは反省したい。

3.4 StudyTyping

文責: 北村 優奈

3.4.1 概要

簡単なタイピングゲームである。内容は出される問題に対して答えを打ち込んでいき、制限時間内にどれだけ回答出来るかというものである。

3.4.2 作品の紹介

スクリプトの生成時に最初から存在する `Update()` というイベント関数は毎フレーム呼び出される関数であるため、万が一 1 フレーム辺りに 2 回以上の入力があると 2 回目以降のキー入力を反映させられない。そのため、キーボードやマウスのクリックの際に呼び出される `OnGUI()` というイベント関数を使用した。

また、複数の方法で入力出来る機能をスクリプトによって実装した [1]。そのため、例えばひらがなの「し」であれば、「si」と「shi」の 2 通りの入力が可能になっている。図 8～図 10 に、作品のスクリーンショットを示す。

はじめる

図 8: スタート画面

制限時間 : 13
総タイプ数 : 21
ミスタイプ数 : 1

個々の機器の障害が起こる確率を下げて、全体として信頼性をあげる考え方の設計手法をなんというか。

fo-ruse-fu

図 9: ゲーム画面

総タイプ数：119

ミスタイプ数：6

もう一度

図 10: リザルト画面

参考文献

- [1] 柔軟な入力方法に対応したタイピングゲームの作り方 <https://crieit.net/posts/dc0eb0f0a74f6e15e362e1f384d28f0d#%E6%9F%94%E8%BB%9F%E3%81%AA%E5%85%A5%E5%8A%9B%E6%96%B9%E6%B3%95%E3%81%AB%E5%AF%BE%E5%BF%9C%E3%81%95%E3%81%9B%E3%82%88%E3%81%86%E5%86%EF%BC%81>

3.4.3 展望

この作品でタイピングゲームの基礎が作成出来たため、ここから発展させて音声の追加や見た目の変更、スコアの計算など、より工夫してオリジナリティのあるものに変えていきたい。

3.4.4 感想

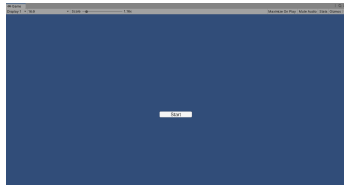
制作したきっかけは1度タイピングゲームを作成してみたいというものだったので、最低限の目標は達成出来たと思う。制作に当たってはUnityにおいてキーの入力がどのように処理されるかなど新しい知見が得ることが出来、とても良い経験になった。

3.5 Mysterious Fortification

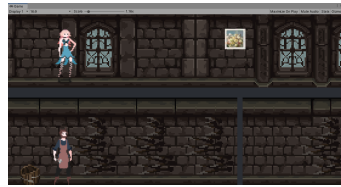
文責: 桐井 優実

3.5.1 概要

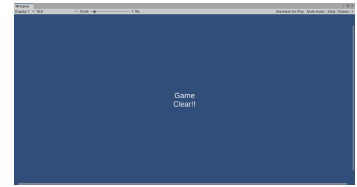
仕掛けをとりて城から脱出する、2Dの謎解きゲームである。最大の特徴は2人のキャラを切り替えながら交互に仕掛けをとくことで先に進むシステムである。



[1] スタート画面



[2] ゲーム画面



[3] クリア画面

図 11: ゲームの流れ

3.5.2 作品の紹介

このゲームはスタート画面、ゲーム画面、クリア画面の3つのシーンから成る。スタートボタンをクリックすることでゲーム画面に遷移し、特定の条件を満たすことで出現する出口から脱出することでクリア画面に遷移する。

ゲーム画面では、上のキャラを動かしたい時はHキー、下のキャラを動かしたい時はLキーを押すことでキャラの操作とカメラの視点を切り替えることができる。

カメラの切り替えにはCinemaSceneのVirtualCameraを利用している。VirtualCameraはPriorityという値をもち、この値がもっとも高いVirtualCameraがゲーム画面として表示されるカメラになる。そのため、「キャラを切り替えるキーを押したらPriorityを変化させる」という動作をする図12のようなスクリプトをEmptyObjectにアタッチすることで、2つのVirtualCameraがそれぞれ上のキャラと下のキャラを追いかけるように設定した。

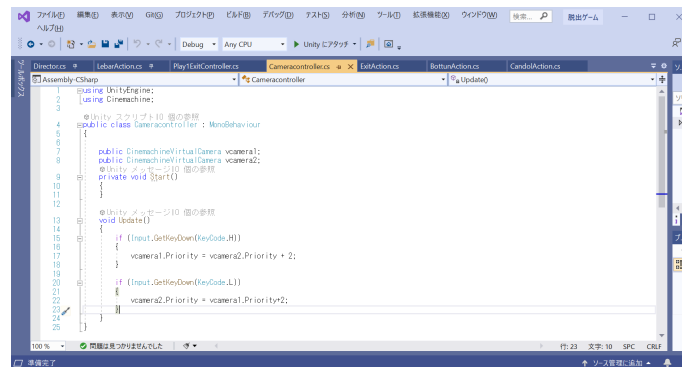


図 12: カメラを制御するソースコード

カメラの移動範囲の設定には、範囲の周辺にPolygonColliderをつけて制御するやり方[1]を採用した。このゲームでは画面上半分と下半分の仕掛けが相互に作用しあい、先に進むことができるようになる。

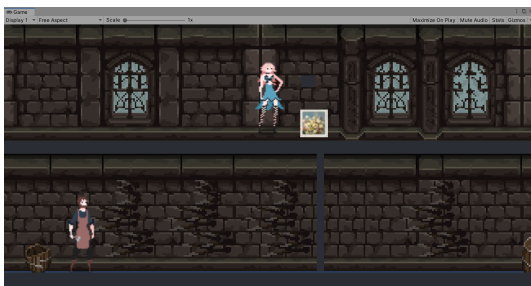


図 13: ボタンを押すと

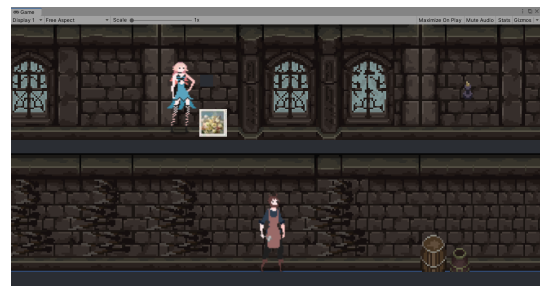


図 14: 下のキャラが進めるようになる

仕掛けの操作を行う仕組みにはisTriggerをオンにしたColliderによる当たり判定を利用している。「キャラがある条件下の元で一定の範囲内に存在するとき、Spaceキーを押すと仕掛けが作動する」というスクリプトを書いた。

void OnTriggerStay2D内で「一定の範囲内に存在するかどうかの判定」を行い、void Update内で「判定が真ならば、Spaceキーを押したとき仕掛けが作動する」という流れになっている。void OnTriggerStay2D内にSpace

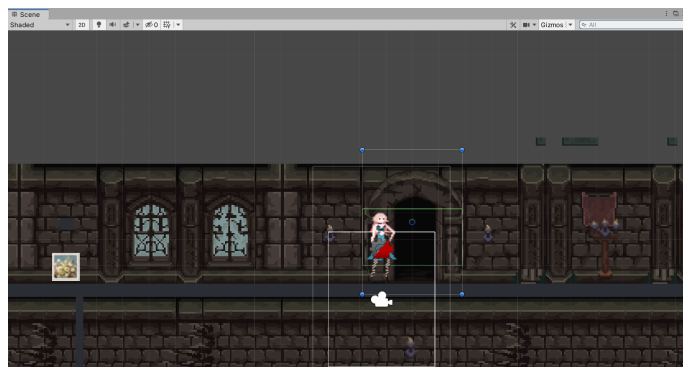


図 15: 仕掛けの仕組み

キーを押したときの動作を直接書くとうまく動作しないため、void OnTriggerStay2D 内では判定のみを行うようにするという工夫をした。

背景はアセットストアから、Tilemap 用の画像を入手して設定した [2].

参考文献

[1] 【Unity】Cinemachine で 2D カメラの表示範囲を制限する <https://nekojara.city/unity-cinemachine-2d-limit>

[2] 【Unity】タイルマップの基本的な使い方 <https://baba-s.hatenablog.com/entry/2018/04/08/130000>

【使用したアセット】

キャラクター Fantasy Environment and Animated Characters(Pixel Art) [free add-on]

背景 Rogue Fantasy Castle

3.5.3 展望

スタート・仕掛けをとく・ゲームクリアという、ゲームの流れは完成させることができた。ゲーム体験を高めるものとして、仕掛けがとけたときの効果音の追加などが考えられる。ステージの追加も検討したい。

3.5.4 感想

Unity を学び始めたのは大学生になってからであるが、今回初めて自分の力で企画・実装・ゲームとしての完成まで持って行くことができた。Unity の理解が大きく進んだのはもちろん、今まで少しずつ蓄積されてきた知識がようやく実践に結びついて理解に繋がった感触があり、今回の活動が自分にとっての大きな一歩になったことを感じた。

4 活動の上での問題点

文責: 山本 京介

活動の問題点として、スケジュールに大幅な遅れが出た事が挙げられる。他のタスクが溜まっている班員が多かったのもあるが、一部では、班員同士の意見交換やアドバイスが円滑に行えず、ゲーム制作が滞ってしまった事が原因であった。出席率の芳しくない班員において、定期的な進捗確認が行えなかった事も原因である。

5 活動で得られた知見

文責: 桐井 優実

Unity に関する知識を持った班員が知識の共有をすることで、班全体の Unity に関する知識量が増えた。また個人制作でゲームの企画から実装までを行うことにより、経験を積むことができた。班員の Unity 経験年数は様々だったが、最後まで参加した班員にとっては実りある経験になったといえる。

6 班総括

文責: 新藤 尚輝

このプロジェクトでは、個人またはグループでゲームを作成していた。そのため、「活動上での問題点」と「活動で得られた知見」の間にあるように、参加している班員の間で乖離が生じた。このような事態に対応しきれなかったのは、反省点である。一方、各々の作品において、やりたいことを実現できていたものが多数であった。できていなかったとしても、何らかの形で学びを得ることができていた。ここから、「活動で得られた知見」にあるように、最後まで参加した班員にとって、実のある経験ができていたと言える。そのため、「Unity の知識を深め、プロダクトを作る」という目標は、おおむね達成できた。