



ポートフォリオ

大阪電気通信大学 総合情報学部 デジタルゲーム学科
中道和希

さくしもち
..11..

目次

自己紹介

└ 自己紹介	3p
└ 保有資格・受賞歴	3p
└ 主に使用しているソフト・言語	3p
└ 趣味・特技	3p

ゲーム

└ NyaX'22	4p
└ ブロックマップ	5p
└ サウンド組み込み	6p
└ タイトルロゴ・デザイン	7p
└ 作曲	7p
└ 制作したツール	8p
└ スケジュール管理BOT	8p
└ Google Drive探索BOT	9p

CG

└ Perfect Crime	10p
└ 3DCG作品	11p

サウンド

└ Octogether	12p
└ 曲解説	13p

その他

└ ゲームジャム運営	14p
└ イラスト作品	14p
└ 同人活動	15p



自己紹介



名前	中道 和希(Kazuki Nakamichi)
大学	大阪電気通信大学
学部・学科	総合情報学部 デジタルゲーム学科

「できる事はなんでもやる」をモットーに制作に取り組んでいます。メインはプログラミングですが、サウンド制作や3DCG、2DCGなど幅広く活動しています。

保有資格・受賞歴

- ・ITパスポート
- ・Microsoft Office Specialist Associate
- ・Microsoft Office Specialist Access
- ・Illustrator®クリエイター能力認定試験
- ・CGクリエイター検定ベーシック
- ・学長奨励賞
- ・学部長賞

主に使用しているソフト・言語

・C# / Unity

サウンド制御処理を主にしています。最近では、エディタ拡張などを行い、より開発が効率的にできるようにしています。

・git / Sourcetree

チーム制作などで、GitHubやGitLabを用いてバージョン管理しています。

・Cubase 12 Pro

ゲームにBGMやボカロ制作のために使用しています。プラグインや音源も多数保有しています。

趣味・特技

・音楽制作

小学生の頃に、ゲーム音楽の素晴らしさに目覚め音楽制作を始めました。独学で勉強し、今は学内のサウンドに興味がある人に向けた交流や勉強のためのサーバーを開設し、運営もしています。また、友人と同人サークル「Cadence」を開設し、代表を務めています。

・DJ

イベントに参加し、そこで出会ったDJの人に色々教えてもらったことをきっかけに本格的に始めました。大学でも、興味がある人向けに体験会などを実施しました。

Webサイト

このポートフォリオに載せていない作品も多数あります。

NyaX'22

NyaX

自己紹介

ゲーム

CG

サウンド

その他

NyaX'22

スペースを押してねん・ω・ん
escでゲームを開じる

2022 mixtuti

ミスしても大丈夫なブロック崩し
制限時間までに全てのブロックを壊そう。
いつも通り、特別な8bitが舞う
隠しコマンドも実装！！

ゲーム名 NyaX'22

ジャンル ブロック崩し

制作期間 2日

制作人数 1人

担当箇所 全て

使用ツール Unity、GarageBand、Cubase 12 Pro、Magical 8bit Plug 2

使用言語 C#

リリース日 2022/11/18

作品概要

画面下にボールが触れてもミスにならないのが大きな特徴なブロック崩しゲームです。ただし、制限時間以内にブロックを全て崩さないといけないので、バーを使って加速させる必要があります。普通にプレイしていても絶対にクリアできず、うまく加速させないと時間も足りないという少しシビアのゲームとなっています。

アピールポイント

隠しコマンドを打つ事で、段々と現在再生されているBGMがフェードアウトし、次に再生されるBGMがフェードインしてくるスクリプトや、ボールがブロックに当たった時に、回数の応じて少しSEを変えるスクリプトなど、サウンドの組み込みに少し力を入れてみました。

また、ステージ数もすぐに増やしやすくようにプログラムを記述しています。

プレイリンク

[Mac版](#)

[Windows版](#)

[Unityroom版](#)



```

12 // 0:None 1:Red 2:Blue 3:Green 4:Yellow 5:Magenta
13 int[,,,] stageMap = {
14     {
15         { 3, 3, 3, 3, 3},
16         { 2, 0, 2, 0, 2},
17         { 1, 1, 0, 1, 1},
18         { 0, 0, 0, 0, 0}
19     },
20     {
21         { 4, 4, 0, 4, 4},
22         { 3, 0, 3, 0, 3},
23         { 2, 2, 2, 2, 2},
24         { 0, 1, 1, 1, 0},
25     },
26     {
27         { 4, 5, 4, 5, 4},
28         { 5, 3, 3, 3, 5},
29         { 2, 5, 2, 5, 2},
30         { 1, 1, 5, 1, 1},
31     }
32 };
33

```

ブロックマップ

上のスクリプトは、0から5の数字によって、ブロックの色を選択してマップを作成しています。その配列の数によって、右上の配置の部分でプレハブを設置していきます。マップの配列部分を変えるだけで、変更や追加などしやすいように設計しました。

```

39 // ステージ作成
40 public void CreateStage(int stage){
41     var parent = this.transform;
42
43     // 配置する座標を設定
44     Vector3 placePosition = new Vector3(-4,0,9);
45     // 初期化する座標を設定
46     Vector3 initPosition = placePosition;
47
48     // 配置する回転角を設定
49     Quaternion q = new Quaternion();
50     q = Quaternion.identity;
51
52     // ブロック全削除
53     var clones = GameObject.FindGameObjectsWithTag ("Block");
54     foreach (var clone in clones){
55         Destroy(clone);
56     }
57
58     // 配置
59     for (int i = 0; i < stageMap.GetLength(1); i++) {
60         placePosition.x = initPosition.x;
61         for (int j = 0; j < stageMap.GetLength(2); j++) {
62             int item = stageMap[stage-1, i, j];
63             if(item != 0){
64                 // ブロックの複製
65                 GameObject block = Instantiate(blockPrefab, placePosition, q, parent);
66                 // ステージのブロック番号により色変更
67                 Renderer r = block.GetComponent<Renderer>();
68                 r.material.color = itemColor[item];
69                 // ステージのブロック番号で名前を生成
70                 block.name = "Block_" + item.ToString();
71             }
72             placePosition.x += blockPrefab.transform.localScale.x;
73         }
74         placePosition.z -= blockPrefab.transform.localScale.z * 2;
75     }

```

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using AudioManager;
5
6  public class PlayBGM : MonoBehaviour
7  {
8      void Start()
9      {
10         BGMManger.Instance.Play(BGMPath.BGM_BLOCK);
11     }
12
13     void Update()
14     {
15         if(Input.GetKey(KeyCode.N)){
16             BGMSwitcher.FadeOutAndFadeIn(BGMPath.NYAN100_CARNIVAL);
17         }
18         if(Input.GetKey(KeyCode.O)){
19             BGMSwitcher.FadeOutAndFadeIn(BGMPath.BGM_BLOCK);
20         }
21     }
22 }

```

サウンド制御

上のスクリプトは、サウンド制御用に作成したAudioManagerを使用して制御を行なっています。スタートした時に、通常BGMを流します。Updateのところでは、指定したキーが押されたときに、現在流れているBGMをフェードアウトさせ、指定したBGMをフェードインさせるというようにプログラムしました。

ソースコード

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using AudioManager;
5
6  public class BlockHitSound : MonoBehaviour
7  {
8      int count = 0;
9
10     void OnCollisionEnter(Collision collision) {
11         if (collision.gameObject.tag == "Block") {
12             if(count % 2 == 0){
13                 SEManager.Instance.Play(SEPath.HIT,0.5f);
14             }
15             else{
16                 SEManager.Instance.Play(SEPath.HIT_ANOTHER,0.5f);
17             }
18             count++;
19         }
20         else if(collision.gameObject.tag == "Wall"){
21             SEManager.Instance.Play(SEPath.WALL_HIT,0.2f);
22         }
23         else if(collision.gameObject.tag == "Racket"){
24             SEManager.Instance.Play(SEPath.HIT_RACKET,0.7f);
25         }
26     }
27 }

```

効果音制御

上のスクリプトは、それぞれタグ制御しており、触れたタグによって音を変えています。また、ブロックに触れた回数によっても音を変えています。

ソースコード

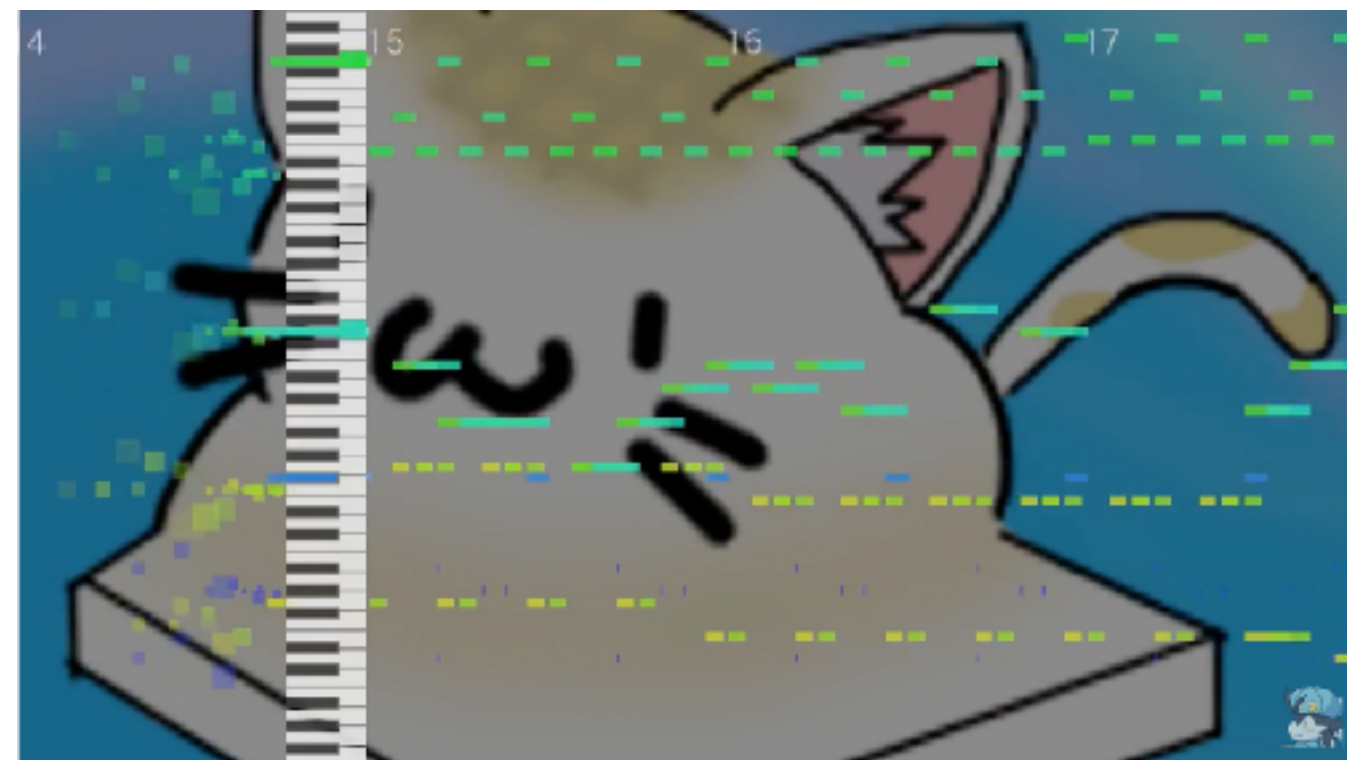
NyaX '22



NyaX

タイトルロゴ・画面デザイン

タイトルやタイトルロゴは、元ネタとしたゲームに雰囲気はかなり寄せて作りました。また、ページ下のステージのデザインも元ネタゲームを参考にして作成しました。Unityに入れる際は、ポストプロセッシングの機能を使い、少し画面をキラキラさせる演出を加えました。



サウンド制作

ゲームのBGMについては、カービィのブロックボールというゲームのBGMを参考に制作をしてみました。画像でいう、高音域の部分の音の動きや、低音域の音の動きの部分です。また、全体的にレトロ感が出るように、8bit風の音を採用し、擬似ディレイなどの手法も取り入れました。また、コード進行は王道進行(V→V→IIIIm→VIIm)を採用しており、長時間のプレイになっても飽きないように、また、少しでも耳に残りやすいように工夫して制作しました。

動画や音楽

[音楽試聴](#)[プレイ動画](#)

制作したツール



自己紹介

ゲーム

CG

サウンド

その他

```
11 for ( var i = 0; i < 7; i++ ) {
12
13     dt.setDate(dt.getDate() + 1);
14     message += Utilities.formatDate(dt, 'JST', '* MM/dd(' + weekday[dt.getDay()] + ')') + "\n";
15     var dayText = "";
16     var calendar = CalendarApp.getCalendarById(CALENDAR_ID);
17     var events = calendar.getEventsForDay(dt);
18
19     var calendarName = calendar.getId()
20     if ( calendarName == undefined ) {
21         message += "予定はありません\n\n";
22         continue;
23     }
24
25     var events = calendar.getEventsForDay(dt);
26     if( events.length == 0 ) {
27         message += "予定はありません\n\n";
28         continue;
29     }
30
```

スケジュール管理BOT

制作の予定や締切が可視化されていないと分かりにくいのではないかと考え、今回BOTを制作しました。

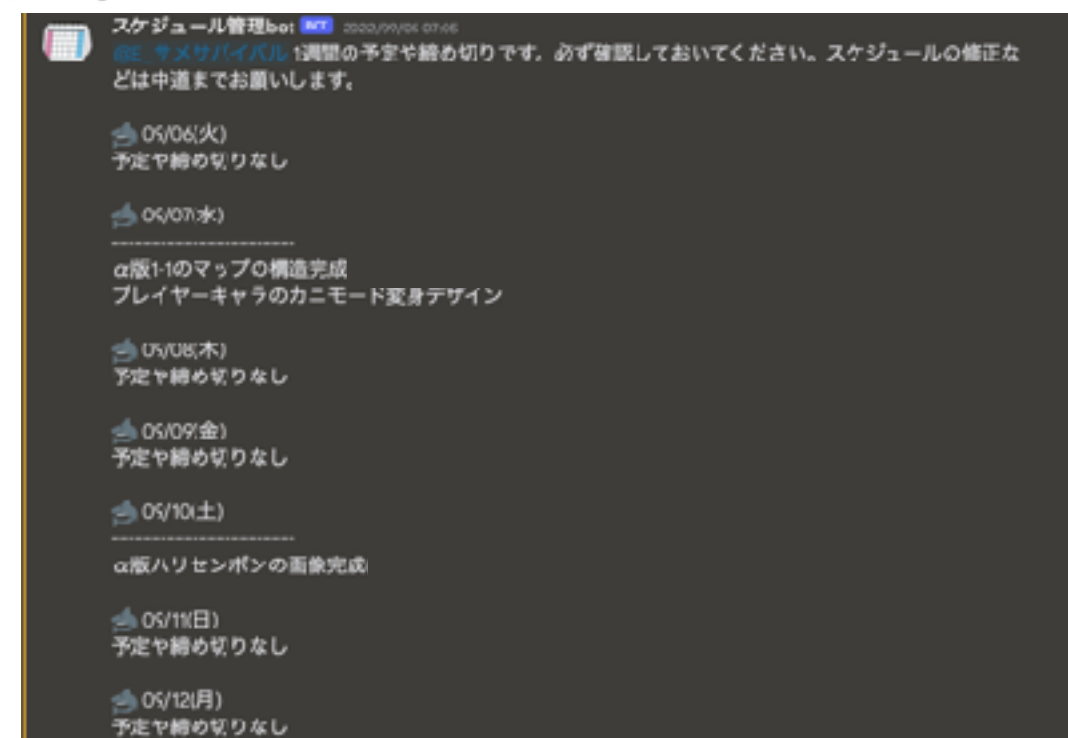
これは、DiscordとGoogle カレンダーを連携させて作りました。7日分の予定を取得し、予定があれば予定を取得し、Discordの指定した場所に出力します。なければ、「予定はありません」などのように指定したテキストを出力します。

上記の処理を毎週決まった時間帯に実行するようにしました。

コード全体

```
31     dayText += "-----" + "\n";
32
33     for ( var j = 0; j < events.length; j++ ) {
34         dayText += String(events[j].getTitle()+'\n');
35     }
36
37     dayText += "\n"
38     message += dayText;
39 }
40
41
42 const payload = {
43     username: "締め切り通知bot(改良版)",
44     content: message,
45 };
46
47 UrlFetchApp.fetch(WEBHOOK_URL, {
48     method: "post",
49     contentType: "application/json",
50     payload: JSON.stringify(payload),
51 });
52 }
```

使用例→



制作したツール



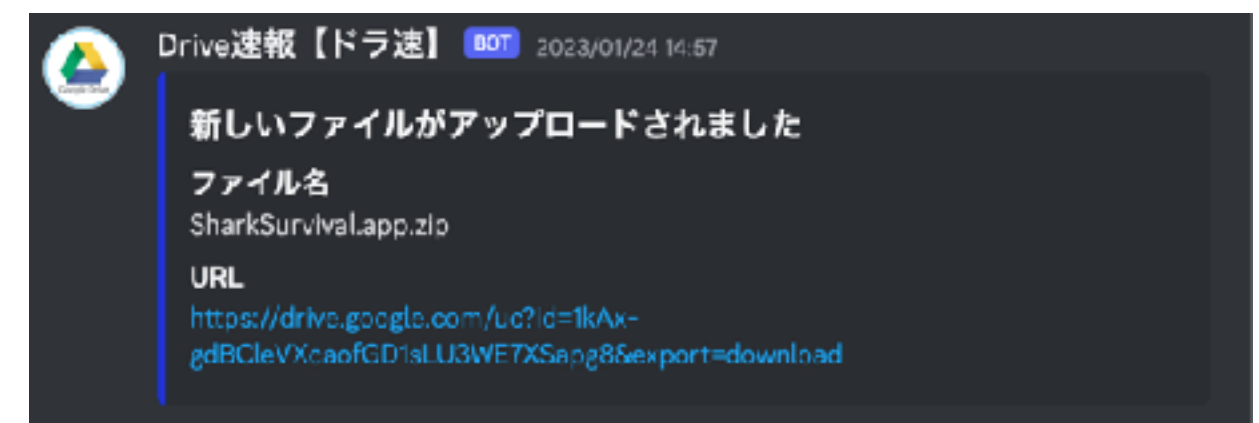
```
13 // 指定フォルダ内の調査
14 function scanFolder(rootFolder, now) {
15
16     // フォルダ内のファイル一覧を取得
17     const files = rootFolder.GetFiles();
18
19     // files内で最近作成されたものを探してDiscordに送信
20     judgeFile(files, now);
21
22     // フォルダ内のフォルダー一覧を取得
23     const folders = rootFolder.getFolders();
24
25     // それぞれのフォルダについてscanFolderを実行
26     while (folders.hasNext()) {
27         const folder = folders.next();
28         scanFolder(folder, now);
29     }
30 }
```

```
33 function judgeFile(files, now) {
34
35     // すべてのファイルについてアップロード時間を判定
36     while (files.hasNext()) {
37         const file = files.next();
38         // ファイルの作成(アップロード)時間を取得
39         const createDate = file.getDateCreated();
40         // 分単位で差分を計算
41         const timeDiff = (now.getTime() - createDate.getTime()) / (60 * 1000);
42
43         // 1分以内に作成されたファイルだったとき
44         if (1 >= timeDiff) {
45             // Log出力
46             Logger.log(file.getName());
47             // Discordに送信
48             sendDiscord(
49                 [{
50                     "title": "新しいファイルがアップロードされました",
51                     "color": 1127128,
52                     "fields":
53                     [{
54                         "name": "ファイル名",
55                         "value": file.getName()
56                     },
57                     {
58                         "name": "URL",
59                         "value": file.getDownloadUrl()
60                     }
61                 ]
62             );
63         }
64     }
65 }
```

Google Drive探索BOT

これは、DiscordとGoogle Driveを連携させて作りました。新しいファイルを作成したときにお知らせしてくれます。制作を進めていく上で、毎回素材の場所を探すのが大変に思い、楽にしたいという思いから制作してみました。URLをクリックすると自動でダウンロードが始まるため、ファイルを探す必要もなく作業に集中することができたと個人的に思っています。上のスクリプトは、特に苦労した、フォルダ内を調査するスクリプトです。左は、最近アップロードされたものかを判定する関数です。もし新しかったら、Discordに埋め込みの形式で送信します。

使用例→



[コード全体](#)

Perfect Crime



ゲーム名	Perfect Crime〜めざせ完全犯罪〜
ジャンル	謎解き脱出ゲーム
制作期間	4日
制作人数	5人(プログラマー:3 グラフィック:1 サウンド:1)
担当箇所	BGM・SE 電灯や机、ベッド、工具箱のモデル 割れたガラスや、包丁などのグラフィック
使用ツール	Unity、MIXTURE BASIC、FL Studio 20、DSPSci-Fi、 blender、MediBang Paint Pro
使用言語	C#
リリース日	2021/08/14

作品概要

様々な証拠となる品物を適切な方法で隠滅を謀るゲームです。途中、2種類のミッションも発生します。このゲームは、「ふれる」というテーマで制作された謎解き脱出ゲーム？です。「法に触れる」から着想を得て制作しました。今回は部屋や電灯などの3Dモデル、窓ガラスの割れた模様や一部の証拠などのグラフィックも担当しました。

アピールポイント

このゲームは、初めてblenderで3Dモデルを作成した作品です。また、3Dモデル制作に留まらず、2Dのアイテム制作や、サウンド制作、プログラムの助言など様々なことに挑戦したゲームとなっています。

プレイリンク・動画

[Unityroom版](#) [動画](#)



Perfect Crime



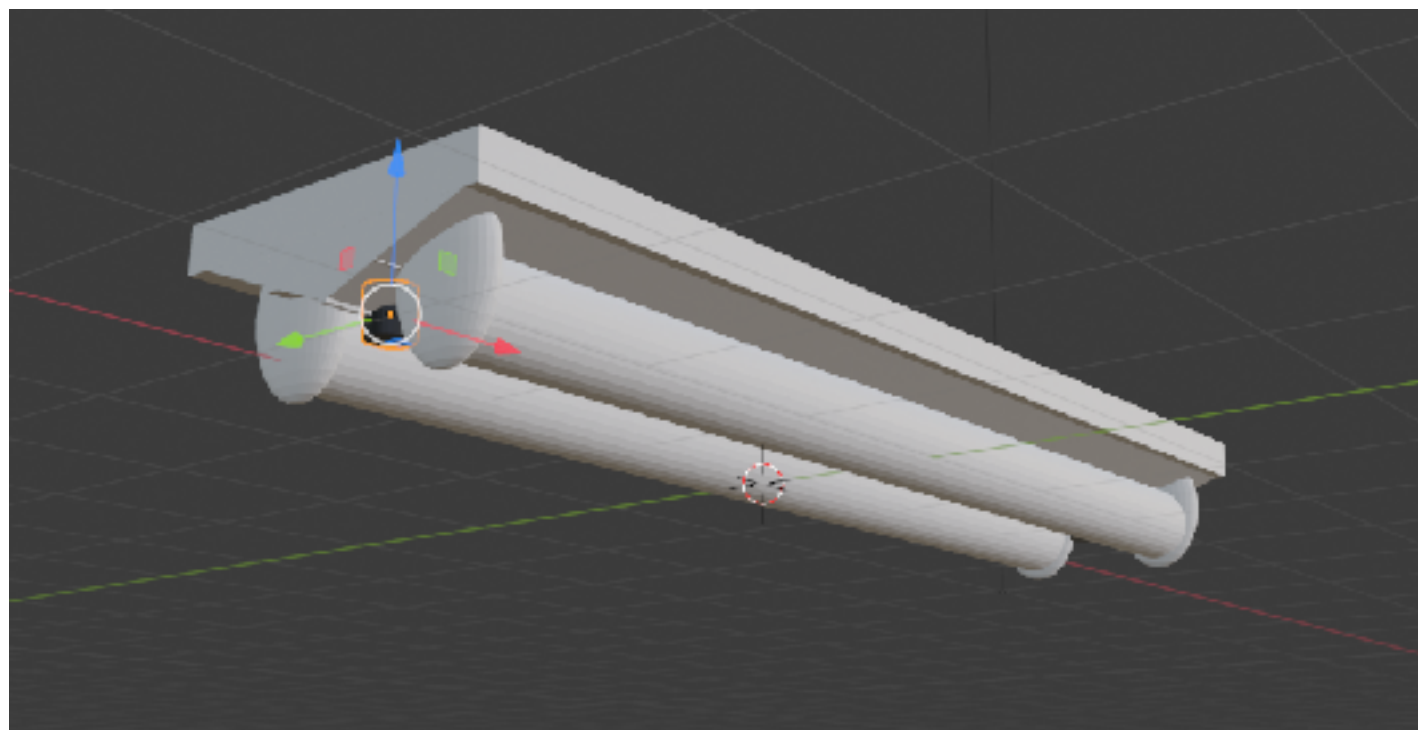
自己紹介

ゲーム

CG

サウンド

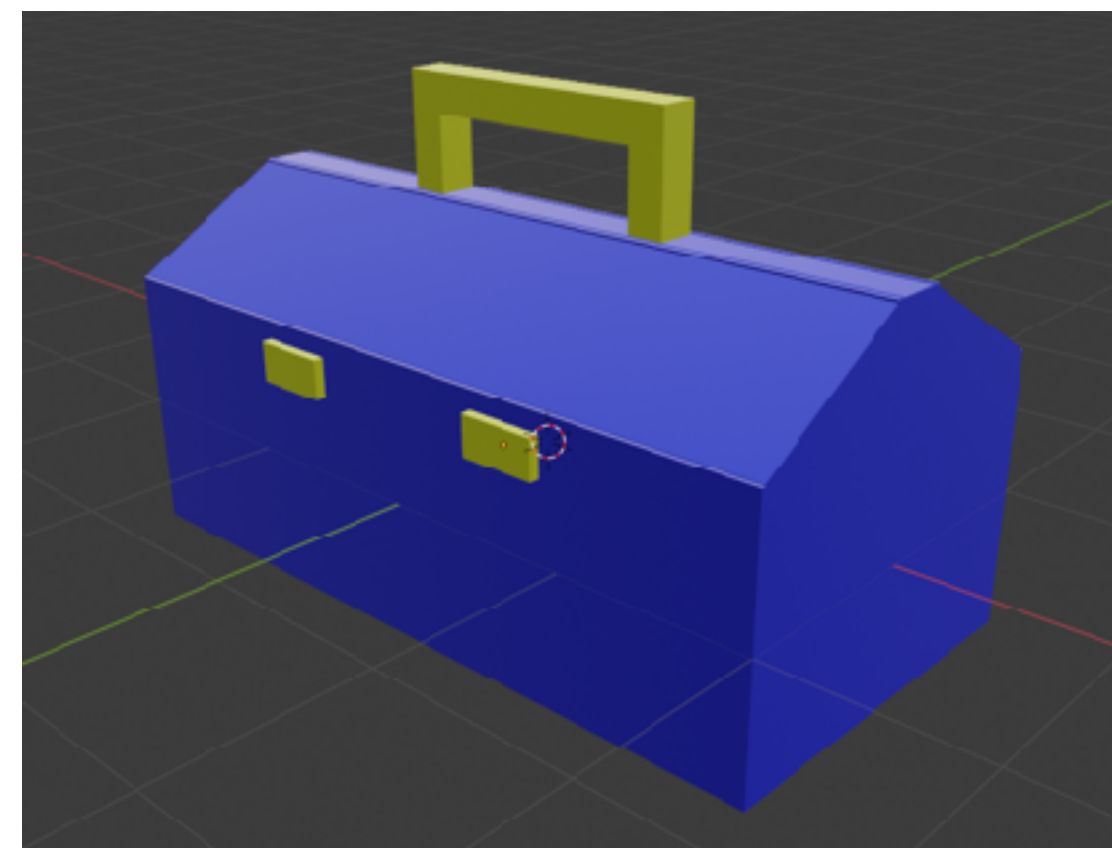
その他



作品概要

こちらは、部屋の照明用に作成した電灯のモデルです。初めてのblenderによるモデル制作だったため、制作には少し手間取りましたが、概ね満足のいく物にできたと思っています。

作品名	電灯
制作期間	2時間
使用ツール	blender



作品概要

こちらは、工具箱のモデルです。証拠品であるトンカチの側に置くために作成しました。

作品名	工具箱
制作期間	1時間
使用ツール	blender

Octogether



作品概要

このゲームは、2021年に開催された「BitSummit Game Jam」にて制作したゲームです。「柔軟性・適応性」というテーマで制作された謎解きゲームです。かなり謎解きの難易度も高いゲームとなっています。

今回自分は、サウンドデザイナーとして参加しました。

アピールポイント

このゲームは、楽曲に統一の主題を持たせてみたり、SE作成用にシンセサイザーを購入したりなどかなりサウンドに力を入れて制作した作品です。

今回は、ビジュアル重視のゲームなため、雰囲気合うようなBGMを作るのにかなり苦労しました。その分、納得できるサウンドを作ることができたのでしました。

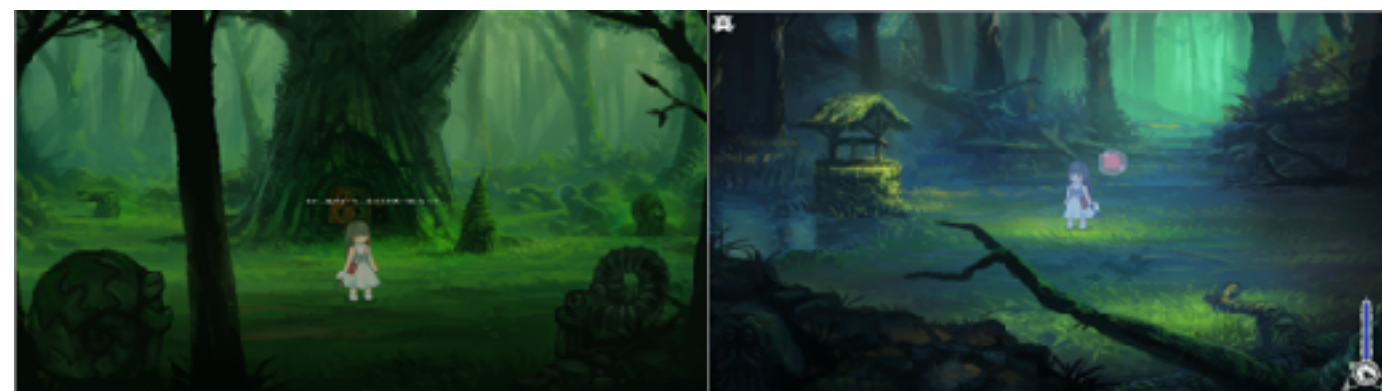
Octogether

ゲーム名	Octogether
ジャンル	2Dアドベンチャーゲーム
制作期間	3日
制作人数	9人(企画:1 プログラマー:3 グラフィック:4 サウンド:1)
担当箇所	BGM・SE
使用ツール	MIXTURE BASIC、FL Studio 20、Domino、DSP fantasy、DSP Anime
使用言語	なし
リリース日	2021/08/08

プレイリンク・動画

[itch.io版](https://itch.io/octogether)

[トレーラー](#)



Octogether



自己紹介

ゲーム

CG

サウンド

その他



曲の解説・動画リンク

この動画は、「Octogether」のゲーム内で使った、もしくは未使用曲を集めました。曲自体は、5曲と実際にゲームで使用したSEを収録しています。ちなみにゲームのタイトル画面は差し替えが間に合わずに少し前のバージョンの曲が使用されています。

曲自体は、ゲームの雰囲気に合わせて全体的に少しエスニック風に仕上げました。また、統一の主題を編曲して使うことで、ゲーム全体の雰囲気の統一を図りました。

[動画](#)

[bandlab](#)

個別の解説・コメント

1. to Mysterious Forest / mixtuti桜餅

タイトル画面のテーマです。未知との出会いということであまり馴染みのない拍子を中心に今回の楽曲を作りました。タイトル曲は4分の6拍子です。笛などの楽器を使い、エスニック風に仕上げました。後で説明しますが、とある事情で曲のメロディを変えました。

2. The Cave / mixtuti桜餅

洞窟のテーマです。正直これが今作でいちばんの自信作です。4度堆積和音などの理論も使い、また「ぴちゃん」というSEも入れてみました。そして、この曲は、4拍子・3拍子・6拍子と可変拍子の曲となっています。

3. Flower Garden / mixtuti桜餅

花園のテーマです。これは結構作るのに苦労しました。7拍子に初挑戦してみたという曲です。

4. In Memory Octogether / mixtuti桜餅

エンディングのテーマです。感動できるようなメロディになればいいなと思いながら作りました。この曲が最初に完成して、このメロディを変えながら全体的な統一感を出そうと思ったのでタイトル曲のメロディも変更しました。この時から、ゲームサウンドに統一的な主題を持たせることにハマりました。ちなみにこの曲は3拍子で作られており、BPMも頻繁に変わるようにしています。

5. Encounters with Unknown / mixtuti桜餅

メンダコのテーマです。どちらかと言うとメンダコと出会うシーン用の曲です。こちらも同じメロディの編曲です。しかし、この曲は唯一4拍子という通常の曲です。ピアノメインの曲にしてみました。

ゲームジャム運営

イラスト作品

自己紹介

ゲーム

CG

サウンド

その他



イベント概要

このイベントは、学内で行われたゲームジャムという短期間でゲームを作成するイベントです。このイベントでは、参加者のサポートの他に、Discordのロールの割り振り、配信時の音声設定周り、効果音提供、記録用映像の撮影、各チームに寄せられていたフィードバックコメントの共有、一部資料作成を行いました。



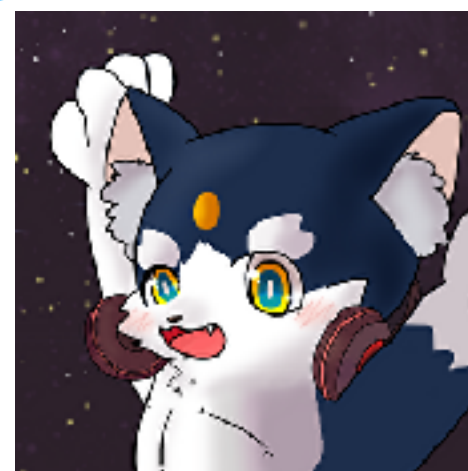
ゲームに出てくる敵キャラを考えるというイベントにて作成しました。洞窟に生息し、毒のブレスを吐くという設定のため、配色は紫をベースに毒々しい色合いにしました。

作品名	レグ・ダーディ
制作期間	6時間
使用ツール	MediBang Paint Pro



このキャラクターは自身が開発したbot用のアイコンとして作成した画像です。猫餅と呼ばれる餅があると知り、こんなだったら可愛いだろうという想像を膨らませながら書いてみました。頭にはお餅が焼けているかのような模様にしてみました。

作品名	モチネコ
制作期間	6時間
使用ツール	MediBang Paint Pro



サークルで出す予定のCDジャケットの案として書きました。テーマが「宇宙ネコ」です。音楽系のサークルなのでその要素も少し入れてデザインしました。

作品名	宇宙ネコ
制作期間	10時間
使用ツール	Photoshop

同人活動



風のクロノア2 20周年を祝うためのアンソロジー企画に参加しました。ありがたいことに150冊以上売れて現在頒布終了しています。ゲーム中で個人的にもっとも印象に残っているゲーム中のシーンである、哀しみの王と対峙している様子を描いてみました。

作品名	哀しみとの対峙
制作期間	90日間(50hくらい?)
使用ツール	MediBang Paint Pro



サークルとして初めて制作したボカロです。ボカコレやなわフェスという学内のイベントに出しました。

テーマ設定や曲調やMVの雰囲気など、3人で話し合いながら制作を進めました。

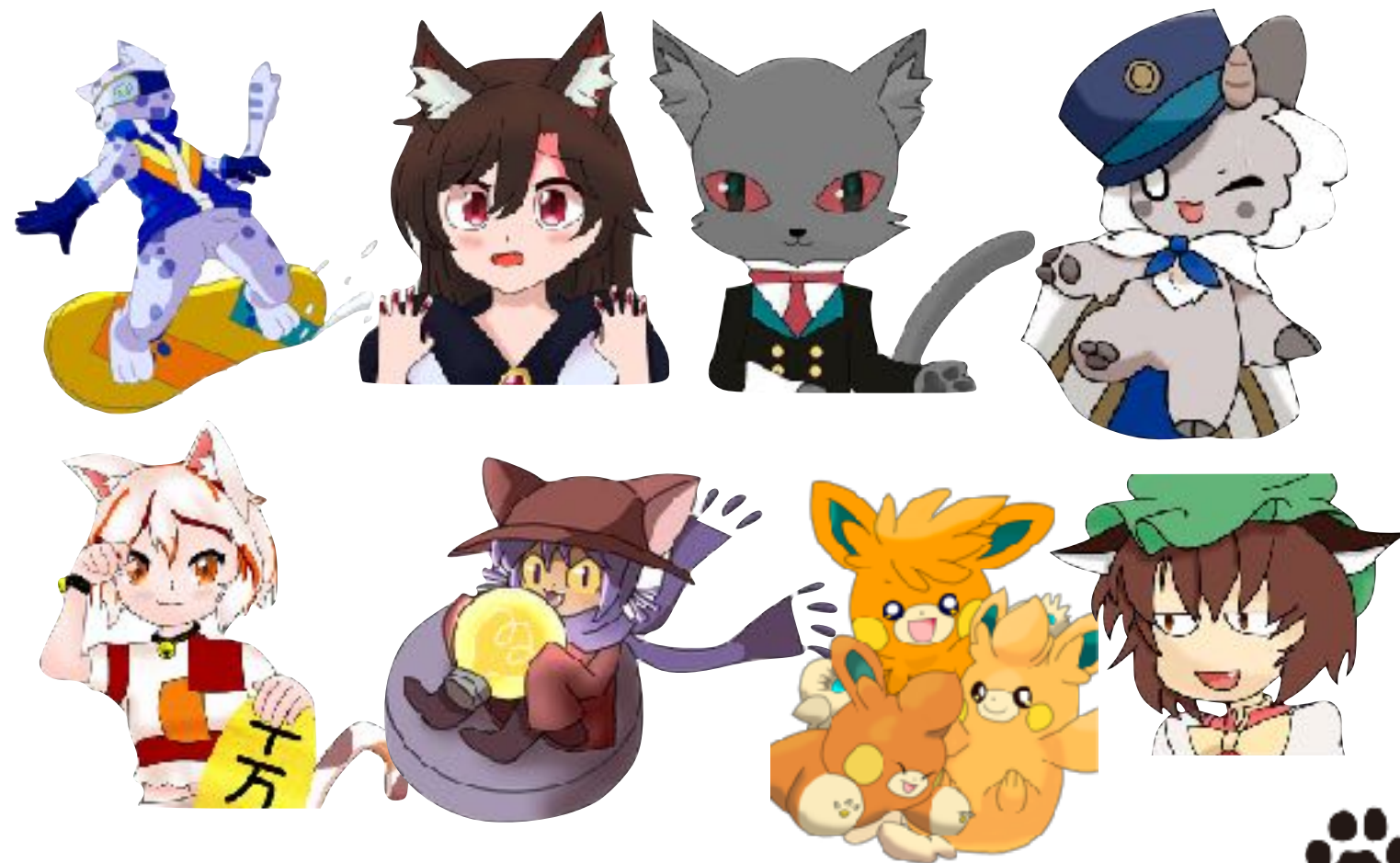
作品名	サイコロタイムジェリー
人数	3人(作詞作曲・MV:1 編曲:1 背景:1)
担当	編曲
制作期間	90日間(50hくらい?)
使用ツール	Cubase 12 Pro



東方旧作未使用曲オンリーのコンピレーションアルバムに参加しました。今回3曲制作しました。前代未聞の旧作未使用曲オンリーの合同アルバムCDとなっています。

作品名	博麗神社境内(trance style mix)
作品名	Dreamy pilot(Hard Style mix)
作品名	IncOmplete Plot
制作期間	それぞれ2日くらい
使用ツール	MIXTURE BASIC、FL Studio

二次創作の落書き



Webサイト

<https://mixtuti.github.io/mixtuti/>

GitHub

<https://github.com/mixtuti>

GitLab

<https://gitlab.com/mixtuti>

YouTube(アレンジ)

https://www.youtube.com/@mixtuti_mochi

YouTube(オリジナル)

<https://www.youtube.com/@Klokirb>

pixiv

<https://www.pixiv.net/users/66647210>

