

BLACKJACK



篠崎 大生

目次



- ❧ テーマ
- ❧ ゲーム説明(1/2)
- ❧ ゲーム説明(2/2)
- ❧ ゲーム画面
- ❧ 独自性
- ❧ プログラミング技術(1/3)
- ❧ プログラミング技術(2/3)
- ❧ プログラミング技術(3/3)
- ❧ クラス図(1/2)
- ❧ クラス図(2/2)
- ❧ 制作過程
- ❧ まとめ(1/2)
- ❧ まとめ(2/2)

テーマ



- ☞ 自分なりに吸収できたことを表現できるポートフォリオとして作成する。
- ☞ サンプルコードを使わず、自作で既存のゲームの動きを実現させる。
- ☞ 上記をもとに制作したWindowsフォームアプリの「BLACKJACK」を、Unityに移植する。

ゲーム説明(1/2)

ブラックジャックの基本ルール



- ❧ ディーラーと1対1で、いかに手札の数字を21に近い数字にするかで勝敗が決まる。
- ❧ Aは1又は11として数え、絵札(J,Q,K)は全て10として数える。
- ❧ 最初にBETする金額を決め、2枚のカードが配られる。その時点で21 (Aと10又は、絵札の組み合わせ)だと、一番強い役 (BLACKJACK)となる。
- ❧ 配当金は、BET額 ×
勝利：2倍 引き分け：1倍 敗北：0
ブラックジャックで勝利すると2.5倍

ゲーム説明(2/2)

本アプリのゲーム仕様



♣ プレイヤーの行動

- HIT(カードを引く)
- SURRENDER(ゲームを降り、BET額の半分が返金)
- STAND(自分のターンを終える)
- DOUBLEDOWN(BET額を倍にして、カードを一枚引いて、自分のターンを終える)

♣ MONEYが、\$ 10,000を超えたらゲームクリア。

♣ MONEYが、BETの最低額を支払えなくなったらゲームオーバー。

ゲーム画面



独自性



♣ ゲーム性

ゲームクリア後は、ボーナスステージとして勝利する度にBETの最低額が勝利時のBET額に変わる、よりハイリスクハイリターンな設定に変化。

♣ プログラミング

トランプの管理を、ブラックジャック以外のトランプゲームでも使用できるように作成し、汎用性を持たせた。

プログラミング技術(1/3)

プログラミングに用いた手法等



- ✧ トランプの管理、プレイヤー、ディーラーの行動の管理をクラス、継承を用いてオブジェクト指向で設計。
- ✧ 勝敗の管理と、ノーマルステージ、ボーナスステージの管理を列挙型を使うことで、コードを見易くした。

プログラミング技術(2/3)

初期制作のWindowsアプリからUnityへの移植時の改善点



☞ トランプの管理方法の改善点

- Windowsアプリとして制作時 → 数字、トランプの絵
- Unityへの移植時 → 上記に加えスーツ(♠♥♦♣)も対象とした。

☞ プレイヤー、ディーラーの管理方法の改善点

- Windowsアプリとして制作時 → 変数のみ全てを管理
- Unityへの移植時 → 基底クラスを作成し、継承を用いて各々を管理

☞ 勝敗の管理判定方法の改善点

- Windowsアプリとして制作時
 - 一つのif文内に勝敗決定時の行動を全て記述。
- Unityへの移植時
 - メソッドで行動を分け、見易くした。

プログラミング技術(3/3)

初期制作のWindowsアプリからUnityへの移植時の変更点

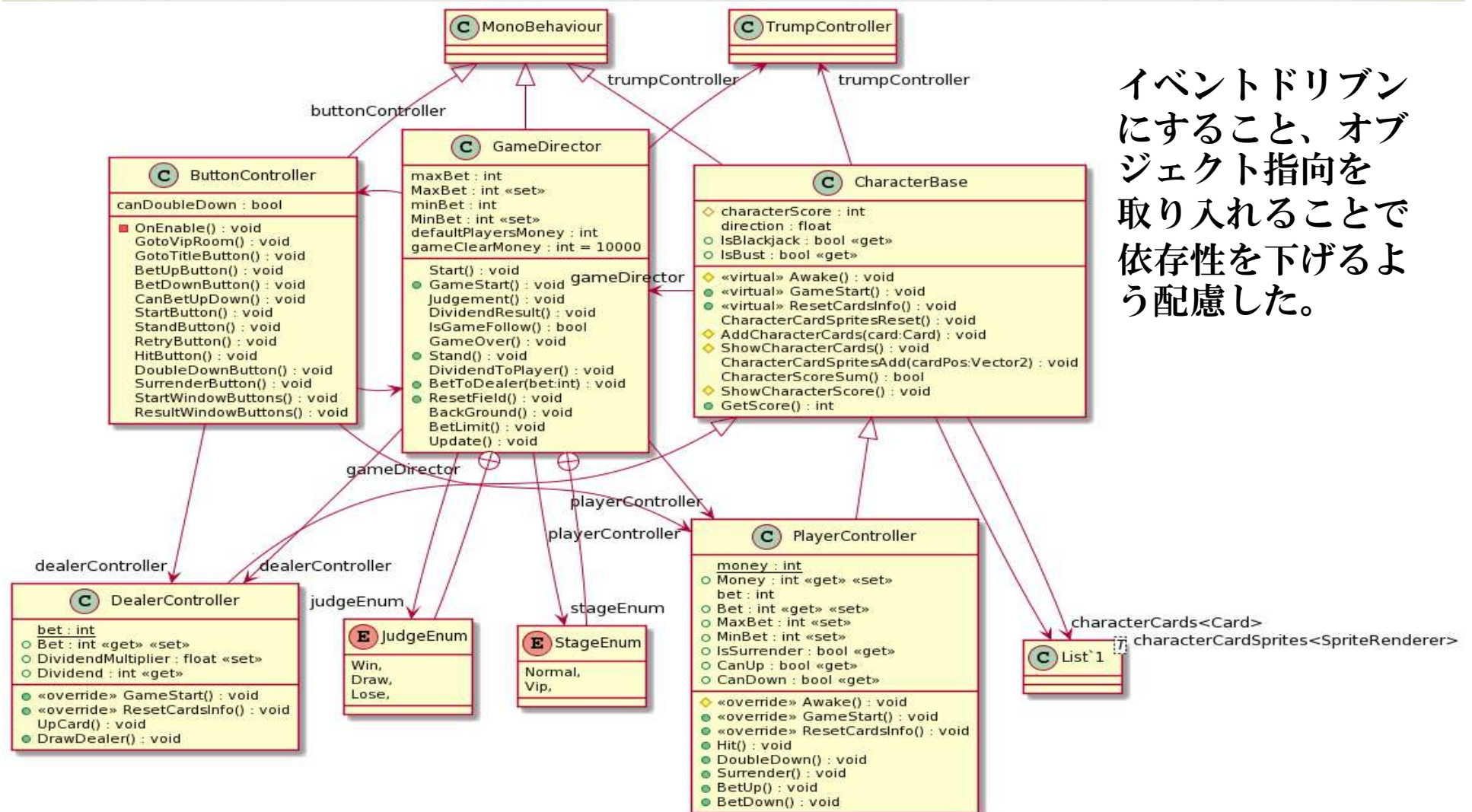


☞ ゲームの独自性の変更点

- Windowsアプリとして制作時
 - ディーラーの性格機能を追加
(性格によってカードを引く数字の値を変化)
- Unityへの移植時
 - VIPRoom機能を追加
(ボーナスステージとしてハイリスクハイリターンな設定に変化)

クラス図(1/2)

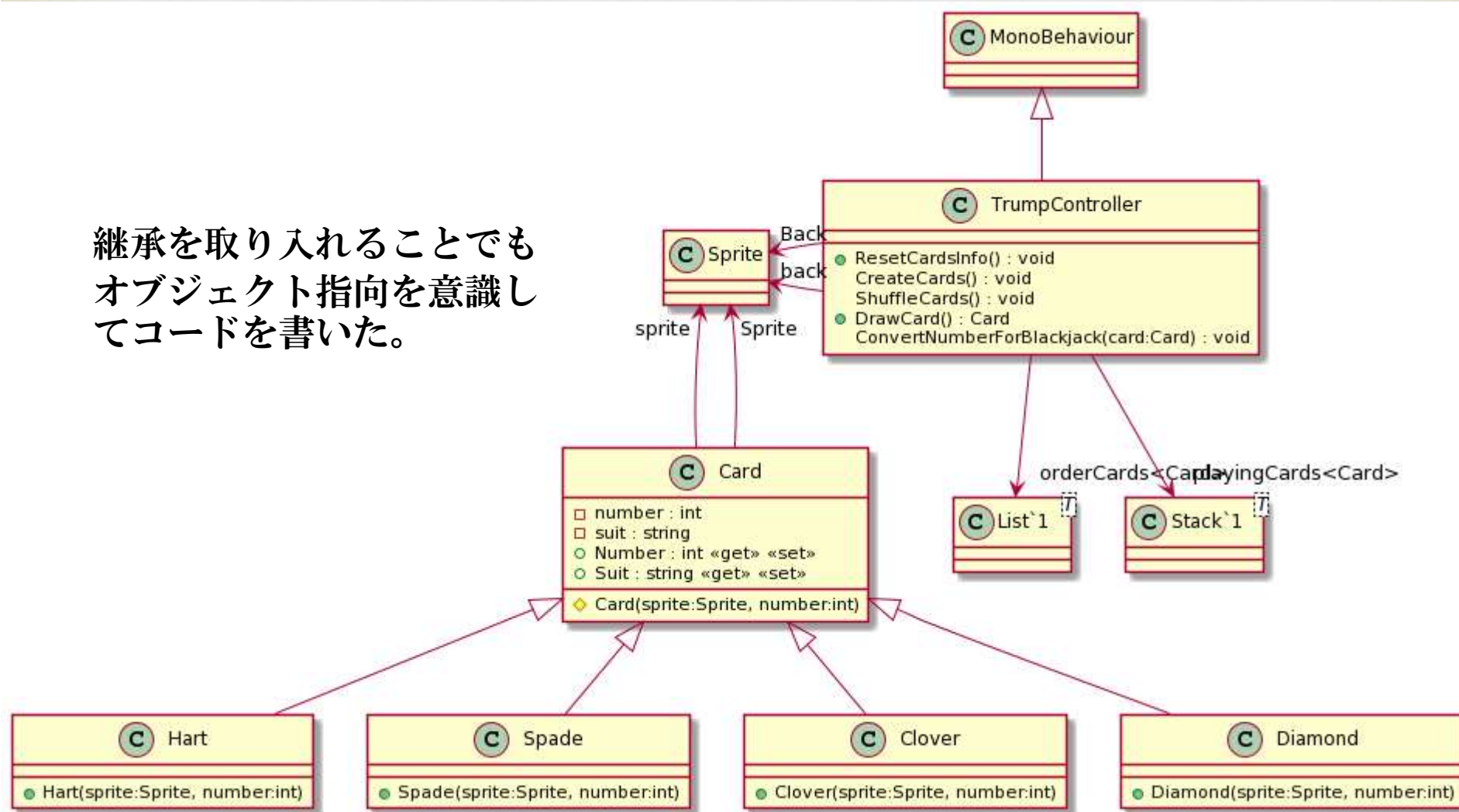
ゲーム管理



クラス図(2/2)

トランプ管理

継承を取り入れることでも
オブジェクト指向を意識し
てコードを書いた。



制作過程



- ❧ メイン画面作成
- ❧ トランプの管理クラス作成
- ❧ カードの配置、表示確認
- ❧ トランプの数字の計算機能追加
- ❧ ボタン管理クラス作成
- ❧ ボタン押下時の各メソッド追加
- ❧ 勝敗判定機能、掛け金機能、UIの追加
- ❧ ゲームクリア、ゲームオーバー条件の追加

まとめ(1/2)



- ❧ 半年前まではパソコンに触れる機会もほぼなく、操作もままならない状態だった。
- ❧ 二か月弱の短期で集中して取り組むことで、Windowsアプリで「BLACKJACK」を作成することが出来るまでになった。
- ❧ ただ、コードにまとまりがなく、機能追加、修正を加えることが難しいものとなってしまった。
- ❧ さらに二か月弱、集中して取り組み、前回の反省も踏まえた上で、上記アプリをUnityへ移植することが出来た。

まとめ(2/2)



- ✧ Unityへの移植を通じて、Unityの理解を深めることが出来、C#のスキルアップにもつながる結果を得られた。
- ✧ また過去の自分のコードを見返し、自分の成長を大いに感じる事が出来、これからの取り組みに対する大きな励みとなった。