

TermProJect GangWonLand

201701994 김재인

201701997 김주영

201702004 김해람

201702015 박영진

1. 프로젝트 개요

1.1 개발 프로그램 명

GangWonLand

1.2 개발 기간

2018.09.18 ~ 2018.12.06

1.3 개발언어 및 개발 환경

C++, Visual Studio Code



2. 프로젝트 개발 모델

Model-View-Controller, **MVC**

Model 

Board, Money, Player

핵심기능수행

View 

AppView

화면 입출력

Control 

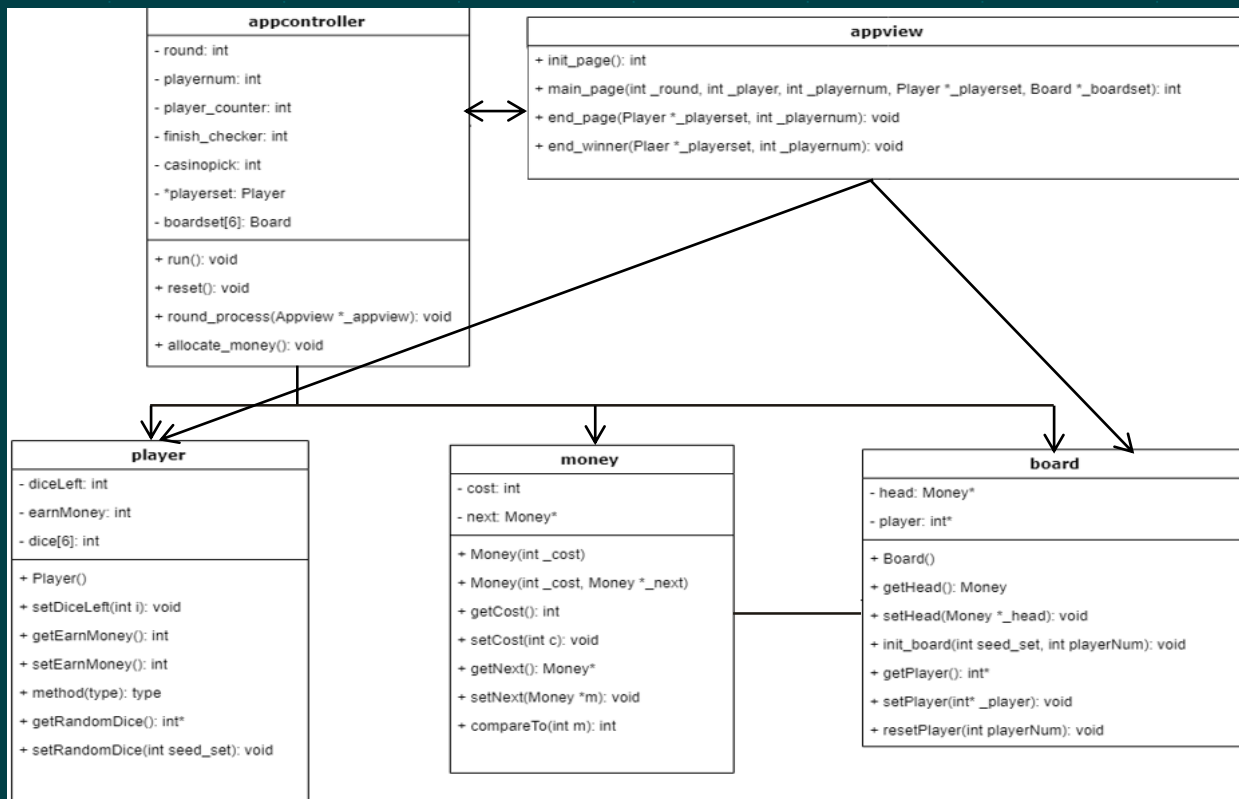
AppController

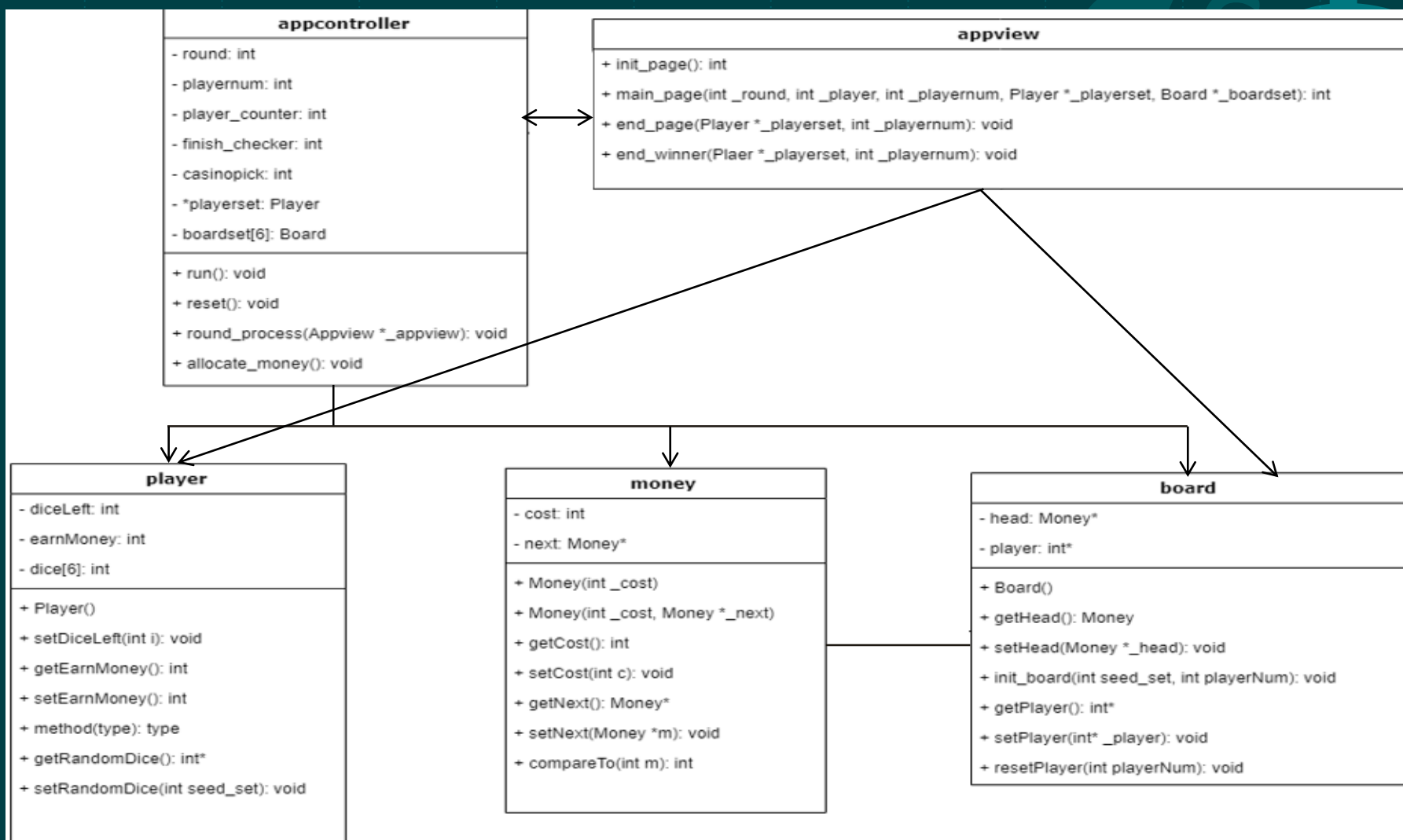
App 전체를 통제

프로그램 시연



3. UML





3.1 AppView

appview

+ init_page(): int

+ main_page(int _round, int _player, int _playernum, Player *_playerset, Board *_boardset): int

+ end_page(Player *_playerset, int _playernum): void

+ end_winner(Plaer *_playerset, int _playernum): void

1. init_page() : 시작화면.
run()에서 호출

```
<GANG WON LAND>
```

```
-Input player number : 5
```

```
!Wrong value. please enter a correct one :
```

```
2
```

3.1 AppView

appview

+ init_page(): int

+ main_page(int _round, int _player, int _playernum, Player *_playerset, Board *_boardset): int

+ end_page(Player *_playerset, int _playernum): void

+ end_winner(Plaer *_playerset, int _playernum): void

2. main_page() : 게임의 메인화면.

Player의 상황에 맞게 보드가 출력된다.


```
+ main_page(int _round, int _player, int _playernum, Player *_playerset, Board *_boardset): int
```

<ROUND 1>

CASINO <1>	CASINO <2>	CASINO <3>	CASINO <4>	CASINO <5>	CASINO <6>
+ WIN MONEY 50000 40000	+ WIN MONEY 50000 40000	+ WIN MONEY 60000 40000	+ WIN MONEY 80000 10000	+ WIN MONEY 60000	+ WIN MONEY 70000
+ BET STATUS PLAYER1: 0 PLAYER2: 0	+ BET STATUS PLAYER1: 0 PLAYER2: 0	+ BET STATUS PLAYER1: 0 PLAYER2: 0	+ BET STATUS PLAYER1: 0 PLAYER2: 0	+ BET STATUS PLAYER1: 0 PLAYER2: 0	+ BET STATUS PLAYER1: 0 PLAYER2: 0
+	+	+	+	+	+

<PLAYER 1>

-Roll the Dice...

-Dice : <1> <2> <3> <4> <5> <6>

-Pcs : 2 1 3 1 0 1

-What casino are you bet to? :

3.1 AppView

appview

```
+ init_page(): int  
+ main_page(int _round, int _player, int _playernum, Player *_playerset, Board *_boardset): int  
+ end_page(Player *_playerset, int _playernum): void  
+ end_winner(Plaer *_playerset, int _playernum): void
```

3. end_page() : 플레이어 수입출력.

Round가 끝날 때 마다 호출한다.

<RESULT>

-PLAYER 1 : 160000\$

-PLAYER 2 : 250000\$

-Press any key to next step.

3.1 AppView

appview

```
+ init_page(): int  
+ main_page(int _round, int _player, int _playernum, Player *_playerset, Board *_boardset): int  
+ end_page(Player *_playerset, int _playernum): void  
+ end_winner(Plaer *_playerset, int _playernum): void
```

4. end_winner() : 승자 출력.

플레이어끼리 비교하여 돈이 제일

많은 플레이어 출력.

<WINNER>

The WINNER is Player 1!!!!
Congratulations ~~!!

-Press any key to exit.

3.2 AppController

1. run() : 게임실행.

같은 클래스의 다른 메소드들
(reset(),round_process(),
allocate_money())이 이용된다.

2. reset() : 데이터 초기화.

run()의 큰 반복문에서 round_process()
의 반복문이 끝날 때마다 호출된다.

appcontroller

```
- round: int  
- playernum: int  
- player_counter: int  
- finish_checker: int  
- casinopick: int  
- *playerset: Player  
- boardset[6]: Board  
  
+ run(): void  
+ reset(): void  
+ round_process(Appview *_appview): void  
+ allocate_money(): void
```

3.2 AppController

3. round_process() : 베팅구현.

run()에서 호출되어 ROUND 한 개를 진행한다.

4. allocate_money() : 돈 분배.

ROUND 하나가 끝났을 때 플레이어가 베팅한 결과를 토대로 플레이어에게 지급

appcontroller

- round: int
- playernum: int
- player_counter: int
- finish_checker: int
- casinopick: int
- *playerset: Player
- boardset[6]: Board

- + run(): void
- + reset(): void
- + round_process(Appview *_appview): void
- + allocate_money(): void

4. 게임 플레이 : 객체설명

1. 플레이어 수 입력

```
<GANG WON LAND>
```

```
-Input player number : 5
```

```
!Wrong value. please enter a correct one :
```

```
2
```

appview

```
+ init_page(): int
```

```
+ main_page(int _round, int _player, int _playernum, Player *_playerset, Board *_boardset): int
```

```
+ end_page(Player *_playerset, int _playernum): void
```

```
+ end_winner(Plaer *_playerset, int _playernum): void
```

2. 보드출력 : 주사위굴리기

<ROUND 1>

CASINO <1>	CASINO <2>	CASINO <3>
+ WIN MONEY 90000 20000	+ WIN MONEY 90000	+ WIN MONEY 60000 40000

BET STATUS	BET STATUS	BET STATUS
PLAYER1 : 0 PLAYER2 : 0	PLAYER1 : 0 PLAYER2 : 0	PLAYER1 : 0 PLAYER2 : 0

<PLAYER 1>

-Roll the Dice...
 -Dice : <1> <2> <3> <4> <5> <6>
 -Pcs : 0 3 1 2 1 1
 -What casino are you bet to? : 2_

player

- diceLeft: int
 - earnMoney: int
 - dice[6]: int

appview

+ init_page(): int
 + main_page(int _round, int _player, int _playernum, Player *_playerset, Board *_boardset): int
 + end_page(Player *_playerset, int _playernum): void
 + end_winner(Player *_playerset, int _playernum): void
 + setRandomDice(int seed_set): void

2. 보드출력 : 베팅할 카지노 선택

<ROUND 1>

CASINO <1>	CASINO <2>	CASINO <3>	CASINO <4>	CASINO <5>	CASINO <6>
+ WIN MONEY	+ WIN MONEY	+ WIN MONEY	+ WIN MONEY	+ WIN MONEY	+ WIN MONEY
90000	90000	60000	80000	80000	80000
20000		40000	20000	40000	20000
			20000		

+ BET STATUS	+ BET STATUS	+ BET STATUS
PLAYER1 : 0	PLAYER1 : 0	PLAYER1 :
PLAYER2 : 0	PLAYER2 : 0	PLAYER2 :

appview

+ init_page(): int

+ main_page(int _round, int _player, int _playernum, Player *_playerset, Board *_boardset): int

+ end_page(Player *_playerset, int _playernum): void

+ end_winner(Plaer *_playerset, int _playernum): void

<PLAYER 1>

-Roll the Dice...

-Dice : <1> <2> <3> <4> <5> <6>

-Pcs : 0 3 1 2 1 1

-What casino are you bet to? : 2_

```
casinopick = _appview.main_page(round+1, player_counter, playernum, playerset, boardset);
```


2. 보드출력 : 베팅할 카지노 선택

<ROUND 1>

CASINO <1>	CASINO <2>	CASINO <3>	CASINO <4>
+ WIN MONEY 90000 20000	+ WIN MONEY 90000	+ WIN MONEY 60000 40000	+ WIN MONEY 80000 20000 20000

BET STATUS	BET STATUS	BET STATUS	BET STATUS
PLAYER1 : 0 PLAYER2 : 0	PLAYER1 : 3 PLAYER2 : 0	PLAYER1 : 0 PLAYER2 : 0	PLAYER1 : 0 PLAYER2 : 0

<PLAYER 2>

-Roll the Dice...

-Dice : <1> <2> <3> <4> <5> <6>

-Pcs : 1 1 2 0 3 1

-What casino are you bet to? :

appcontroller

```
- round: int
- playernum: int
- player_counter: int
- finish_checker: int
- casinopick: int
- *playerset: Player
- boardset[6]: Board
```

```
+ run(): void
+ reset(): void
+ round_process(Appview)
+ allocate_money(): void
```

board

```
- head: Money*
- player: int*

+ Board()
+ getHead(): Money
+ setHead(Money *_head): void
+ init_board(int seed_set, int playerNum): void
```

player

```
- diceLeft: int
- earnMoney: int
- dice[6]: int

+ Player()
+ setDiceLeft(int i): void
+ getEarnMoney(): int
+ setEarnMoney(): int
+ method(type): type
+ getRandomDice(): int*
+ setRandomDice(int seed_set): void
```

```
boardset[casinopick].getPlayer()[player_counter] += playerset[player_counter].getRandomDice()[casinopick];
//save player's betting in collected Casino.
```

3. 결과출력 : 플레이어 돈 출력

<RESULT>

-PLAYER 1 : 160000\$

-PLAYER 2 : 250000\$

-Press any key to next step.

appcontroller

- round: int
- playernum: int
- player_counter: int
- finish_checker: int
- casinopick: int
- *playerset: Player
- boardset[6]: Board
- + run(): void
- + reset(): void
- + round_process(Appview *_appview): void
- + allocate_money(): void

appview

- + init_page(): int
- + main_page(int _round, int _player, int _playernum, Player *_playerset, Board *_boardset): int
- + end_page(Player *_playerset, int _playernum): void
- + end_winner(Plaer *_playerset, int _playernum): void

Thanks!

Any questions?

