

Object Oriented Programming

assignment

Qt\_project

설계 분반: 2 (신동화 교수님)

실습 분반: 1 (신동화 교수님)

학번: 2023202032

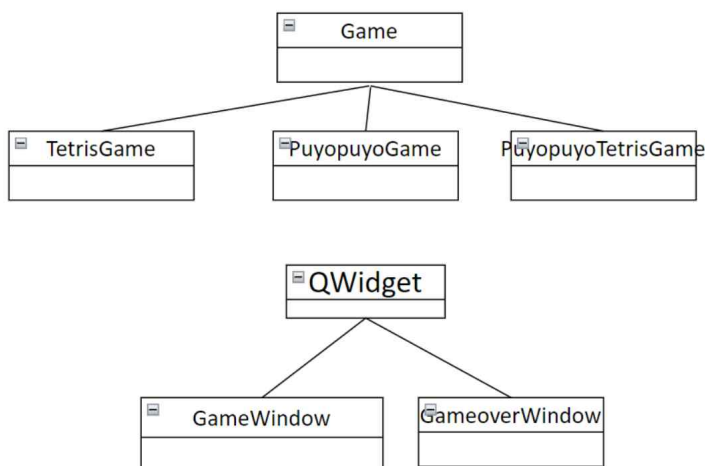
이름: 남호준

# 1. 개요

이번 과제는 뿌요뿌요, 테트리스, 뿌요뿌요테트리스를 구현하는 과제이다.

“ Puyopuyo, Tetris, PuyopuyoTetris를 Qt상에서 모두 구현해야 하며 버전/라이브러리 제약은 없다. 게임들은 개별적인 게임이 아닌, 유사성이 존재하는 게임들이다. 따라서, C++이 가지는 객체 지향 프로그래밍의 세 가지 특성 (encapsulation, inheritance, polymorphism)을 활용한다면 이들을 효율적으로 구현하고 유지보수할 수 있다. 해당 특성들을 활용하여 Puyopuyo, Tetris, PuyopuyoTetris 세 객체를 구현해야 한다. 그리고 객체별 field, method를 어떻게 설계하였고, 객체 지향 프로그래밍의 세 가지 특성을 어떻게 활용하였는지 UML diagram를 그려 보고서에 첨부해야 한다. 세 객체 내부에 또다른 하위 객체가 있어도 되고, 세 객체 외의 또다른 객체가 별개로 있어도 되고, 세 객체를 포함하는 또다른 상위 객체가 있어도 된다. ” 라고한다.

## 2. UML diagram



이 외에도 기본적인 mainwindow, gamewindow 등이 있다.

### 3. class

#### - Game

1) field

int ROW; => 보드 행 길이

int COL; => 보드 열 길이

int LENGTH; => 다음 블록 개수

string\*\* board;//색상 들어가 있는 배열

string\* next;//다음 블록 배열

int front\_id, back\_id;//next 배열의 양 끝 인덱스

int\*\* encoding\_board;//줄 지우거나 뿌요 지울 인코딩 테이블

//0이면 빈칸, -1이면 고정 테트로미노, 1이면 움직이는 테트로미노, 2 3 4 5 6은 차례대로 R, Y, G, B, P를 매핑, 7이면 움직이는 뿌요

Curblock curblock;//생성된 블록 객체

int block[8][4][4] // 블록 모양 정의해놓은 테이블

string combi[32] // 블록 색이나 모양 정의 해놓은 테이블

int score;//점수

int axis\_row;//현재 위치 row

int axis\_col;//현재 위치 col

bool gameover;//게임 오버인지 체크

bool block\_stop;//false면 생성 X, true면 생성하기

random\_device rd;//난수를 위한 변수 선언

2) method

Game(int row, int col, int length);

게임 생성하는 함수

색상 배열, 인코딩 배열을 할당, 초기화함

현재 블록 구조체를 초기화함

배열의

~Game();

게임 오버시 게임 내 할당된 정보를 지우는 함수

## - PuyopuyoGame

1) field

2) method

PuyopuyoGame

뿌요뿌요 형식에 맞게 보드 크기, 다음 블록 개수를 정하고 초기화하는 함수

~PuyopuyoGame

## - TetrisGame

1) field

2) method

TetrisGame

테트리스형식에 맞게 보드 크기, 다음 블록 개수를 정하고 초기화하는 함수

~TetrisGame

## - PuyopuyoTetrisGame

1) field

2) method

PuyopuyoTetrisGame

뿌요뿌요테트리스 형식에 맞게 보드 크기, 다음 블록 개수를 정하고 초기화하는 함수

~PuyopuyoTetrisGame

## - GameoverWindow

1) field

2) method

GameoverWindow

게임 종료 시 안내 문구가 뜨는 팝업 창을 출력함

~GameOverWindow

## - GameWindow

### 1) field

unordered\_map<string, QColor> Colors

=> 색상별 인코딩 해놓은 맵

string block\_shape[7]

=> 블록 모양 배열

int dx[4], int dy[4]=>bfs에서 상하좌우 네 방향 탐색에 사용

int wiro[4]==> 뿌요뿌요테트리스에서 칸에서 얼마나 올라가야 하는지 저장해놓는 배열

int timer=> 시간 재는 변수

random\_device rd=>난수 생성 변수

Game\* p\_game=>게임 클래스 포인터 변수, 게임 보드 등 실질적인 데이터 가짐

QString GAMENAME=> 게임 이름 변수

GameOverWindow\* gameoverwindow=> 게임 오버되면 사용하는 변수

### 2) method

GameWindow(QString gamename, QWidget \*parent = nullptr);

~GameWindow();

void paintEvent(QPaintEvent\* event);

void keyPressEvent(QKeyEvent\* event);

```

void timerEvent(QTimerEvent* event);

void generateblock();//next에 있는 블록을 가져와서 보드에 놓음

void drawNext()블록이 멈추면 블록 생성 후 next에 다음 블록 생성, 아니면 드롭
블록

void dropblock()// //아랫키나 타이머로 drop, soft drop

void movecol(int a)/왼,오 눌렀을 때 이동

void rotate(int a)//0이면 x(시계), 1이면 z(반시계)로 회전

void harddrop()// hard drop

bool blockcorrect(int tmp[4][4], int posX, int posY); // 블록이 멈추도록 하는 함수

void over()/게임 오버 됐을 때 실행되는 함수

void delpreinfo()//delete previous informaion function

void blockplace(Curblock block, int a) //place the block on board

//->a=0이면 테트로미노(p_game 객체 사용), a=1이면 객체 내부 위치 사용

void encoding_update(Curblock block, int a)//블록 고정시 인코딩 정보를 보드에
업데이트

//blockplace 후에 놓인 정보를 바탕으로 encoding됨

void overcheck()//회색 칸에 블록이 있으면 게임 오버

void generate_pu(Curblock* pu)//curblock에서 뿌요 객체 분리

void del_line(테트리스에서 지우는 함수

bool is_valid(int x, int y) 인덱스 유효성 검사 함수

void bfs_pu(int x, int y, vector<vector<bool>>& visited, vector<pair<int, int>>&
to_pop)완탐을 위해 bfs하는 함수

void main_delete_puyo()뿌요 지울 때 호출하는 메인 삭제 함수

```

void pu\_drop()뿌요 지우고 중력 적용하는 함수

bool del\_pu()실질적으로 뿌요 지우는 함수

void drop\_tet\_on\_pu()/테트로미노가 내려가고 내리는 로직 함수

bool pu\_under\_correct()//테트로미노가 내려갈 때 비어있거나 뿌요만 있는지

void swap\_drop\_tet()//뿌요 아래로 테트가 내려가도록 보드를 조정하는 함수

## 4. pseudo code

- 뿌요뿌요

---

### Algorithm 1 Puyo Puyo Game Flow

---

```
1: Initialize game board
2: Generate new Puyo pair
3: while game is not over do
4:   Drop Puyo pair
5:   while Puyo pair is falling do
6:     Wait for player input
7:     if player moves Puyo pair then
8:       Update Puyo position
9:     end if
10:    if Puyo pair lands then
11:      Check for matches
12:      if match found then
13:        Remove matched Puyos
14:        Apply gravity to remaining Puyos
15:        Check for new matches
16:      end if
17:      Generate new Puyo pair
18:    end if
19:  end while
20: end while
21: End game
```

---



## - 테트리스

---

**Algorithm 1** Tetris Game Flow

---

```

1: Initialize game board
2: Generate new Tetromino
3: while game is not over do
4:   Drop Tetromino
5:   while Tetromino is falling do
6:     Wait for player input
7:     if player moves Tetromino then
8:       Update Tetromino position
9:     end if
10:    if Tetromino lands then
11:      Check for completed lines
12:      if line completed then
13:        Remove completed lines
14:        Apply gravity to remaining blocks
15:      end if
16:      Generate new Tetromino
17:    end if
18:  end while
19: end while
20: End game

```

---

## - 뿌요뿌요테트리스

---

**Algorithm 1** Puyo Puyo Tetris Game Flow

---

```

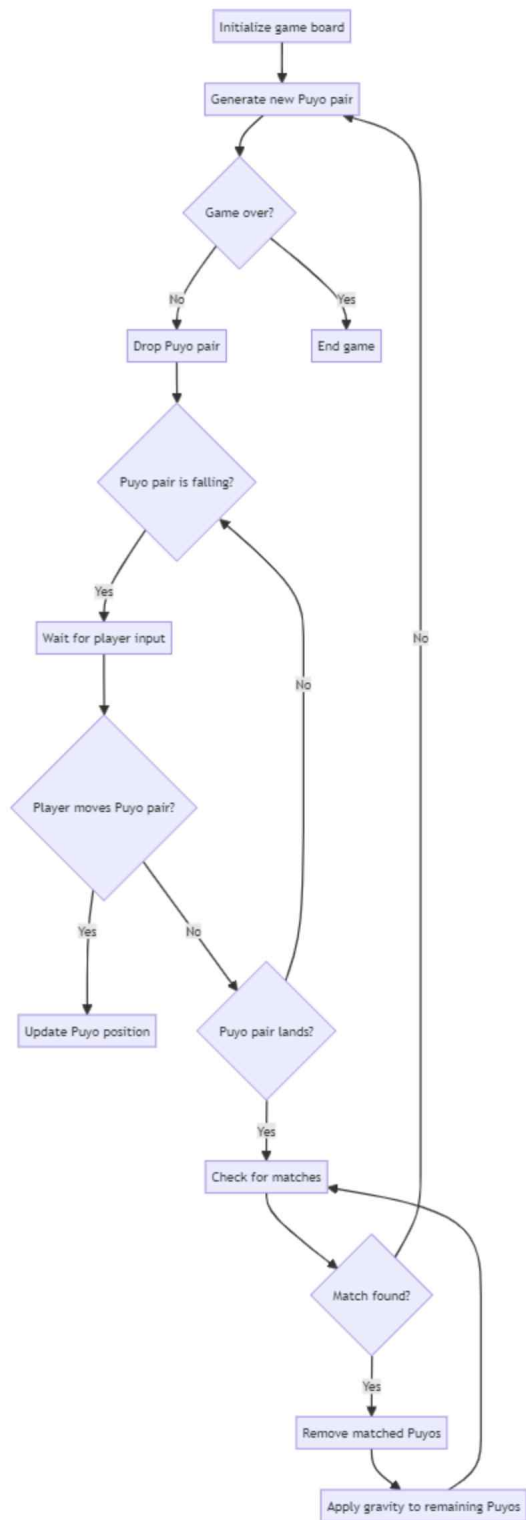
1: Initialize game board
2: while game is not over do
3:   Choose game mode (Puyo Puyo or Tetris)
4:   if mode is Puyo Puyo then
5:     Generate new Puyo pair
6:     while Puyo pair is falling do
7:       Wait for player input
8:       if player moves Puyo pair then
9:         Update Puyo position
10:      end if
11:      if Puyo pair lands then
12:        Check for matches
13:        if match found then
14:          Remove matched Puyos
15:          Apply gravity to remaining Puyos
16:          Check for new matches
17:        end if
18:        Generate new Puyo pair
19:      end if
20:    end while
21:   else
22:     Generate new Tetromino
23:     while Tetromino is falling do
24:       Wait for player input
25:       if player moves Tetromino then
26:         Update Tetromino position
27:       end if
28:       if Tetromino lands then
29:        Check for completed lines
30:        if line completed then
31:          Remove completed lines
32:          Apply gravity to remaining blocks
33:        end if
34:        Generate new Tetromino
35:      end if
36:    end while
37:   end if
38: end while
39: End game

```

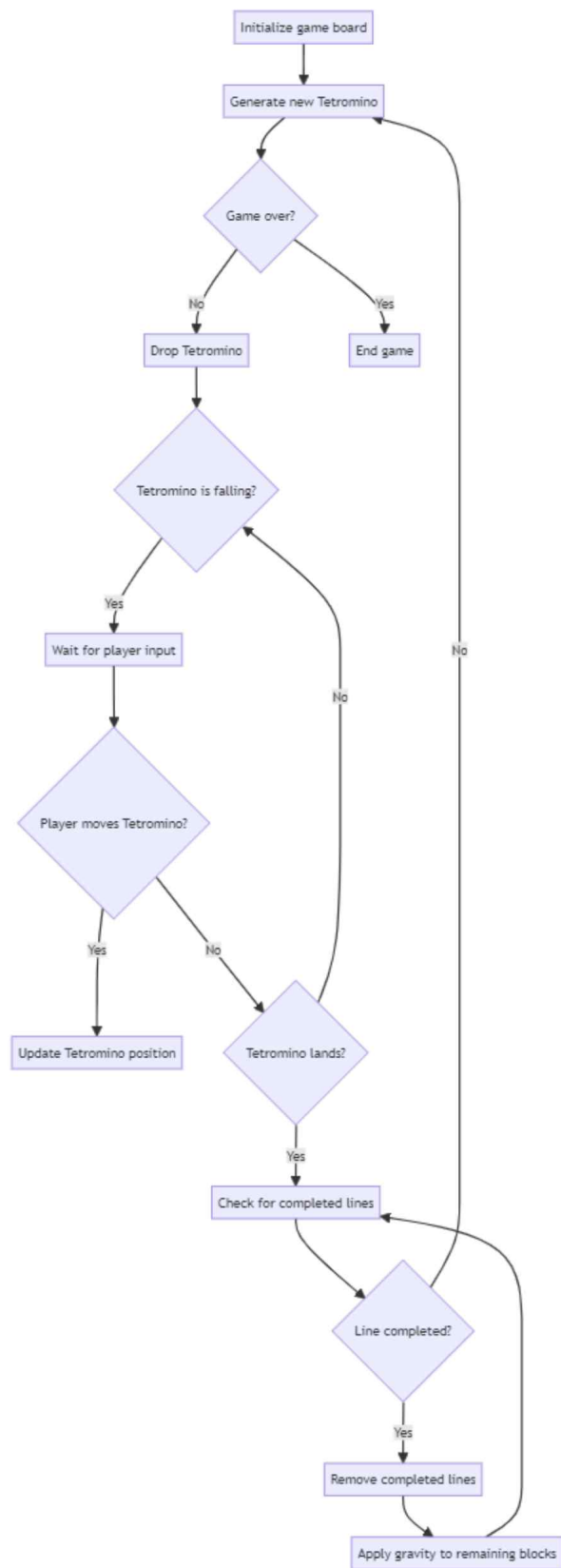
---

## 5. flowchart

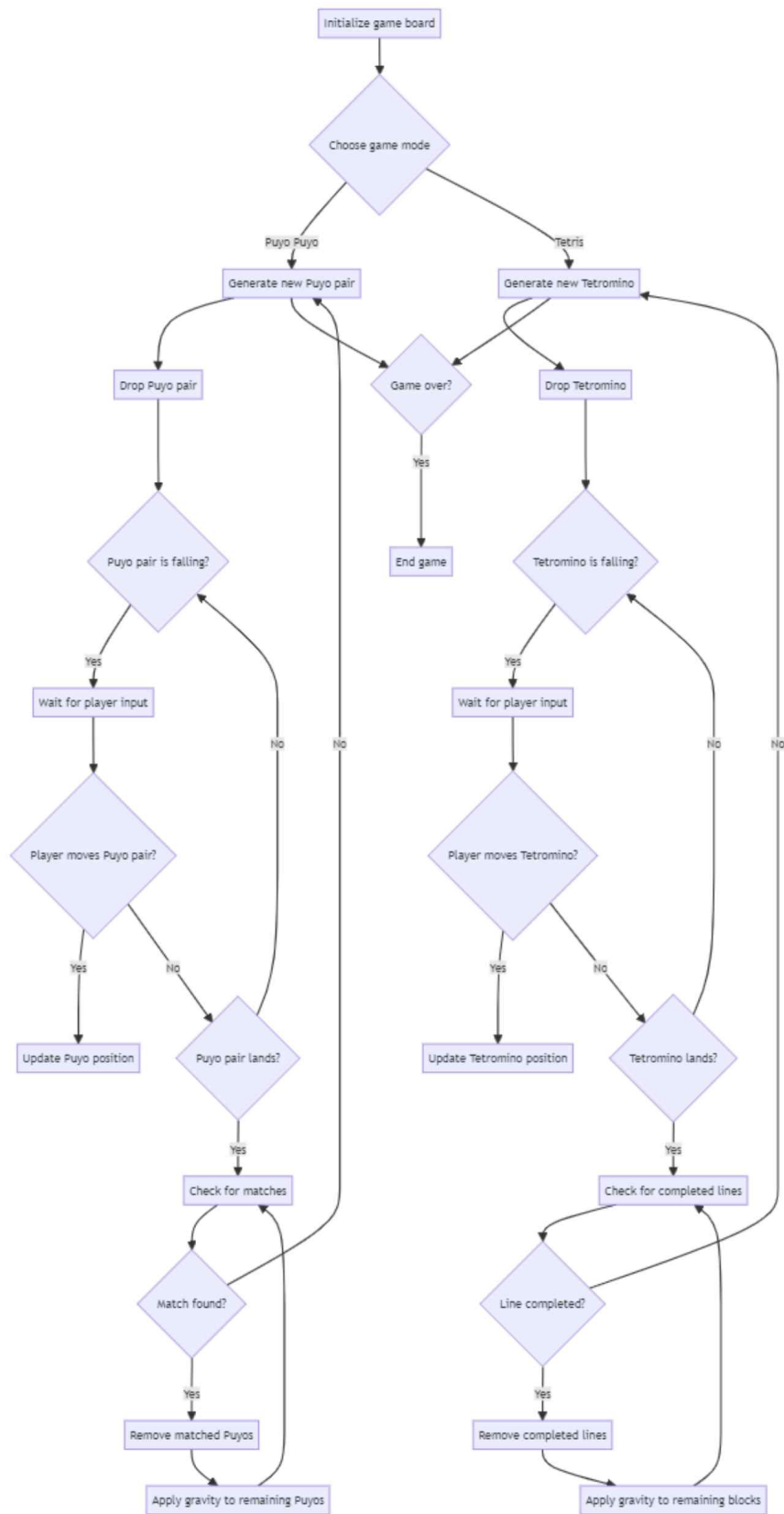
- 뿌요뿌요



## - 테트리스



## - 뿌요뿌요테트리스



timer -  $\frac{2}{2}$  -  $\frac{2}{2}$

~~생성~~ - 생성 - 오버체크 - timer  
오버

~~다옴~~ - 블록맵 체크 - 생성  
드롭

~~드롭~~ - 검사 - 블록 고정  
아래로 푸는

~~위~~ -  $col - 1$  - 검사 - 보드에 넣음 (드롭과 다름)  
최소

~~오~~ -  $col + 1$  - 검사 - 보드에 넣음 (드롭과 다름)  
최소

---

$\frac{2}{2}$  - 바르게 되면 검사 - cur 업데이트 (드롭에서 앞의 세움)  
최소  
보드 업데이트

$\frac{X}{X}$  - 시계방향 검사 - cur 업데이트 (드롭에서 보드 업데이트 앞의 세움)  
최소

$\frac{2}{2}$  - 입계이리 없애기 |  $\frac{2}{2}$  다옴

검사 -

1. 보드 안 바이트까지
2. 블록 제거하기
3. 실제로로 존재해나가

키에 따라서 적은 부분도 있다.

검사(correct)함수들은 보드 인덱스를 넘거나 블록이 겹치지 않거나 인코딩이 되어있거나 다른 추가 조건들로 이루어져있다.

## 6. VIDEO LINK

### - 뿌요뿌요 유튜브

[https://youtu.be/cbKuOLilk\\_c](https://youtu.be/cbKuOLilk_c)

해당 영상에선 단일 chain 삭제, 여러 그룹의 삭제, 연쇄 삭제, 게임 종료 조건인 블록이 위에 닿았을 때 삭제 되도록 하는 영상이 담겨있다.

55초 쯤 노랑색과 빨간색 뿌요쌍으로 한 틱에 두 그룹의 뿌요가 삭제된다.

1분 9초엔 연쇄 작용으로 뿌요가 삭제되는 것을 볼 수있다.

### - 테트리스 유튜브

<https://youtu.be/v5KonE9h7YY>

해당 영상에선 테트리스 영상을 담고 있다.

여러 블록이 회전과 이동을 하는 모습이 담겨있다.

또한, 1 블록으로 한번에 최대 4줄까지 삭제하는 모습이 담겨있다

### - 뿌요뿌요테트리스

<https://youtu.be/EwAURRVh03A>

해당 영상에선 뿌요블록과 테트로미노가 공존하는 게임의 영상이다.

뿌요 블록과 닿게 되면 테트로미노는 맨 아래로 내려가게 된다.

테트로미노는 뿌요블록과 별개로 다르게 적용하여 라인별로 지워지게 되고

뿌요 블록은 뿌요블록대로 삭제되게 된다.