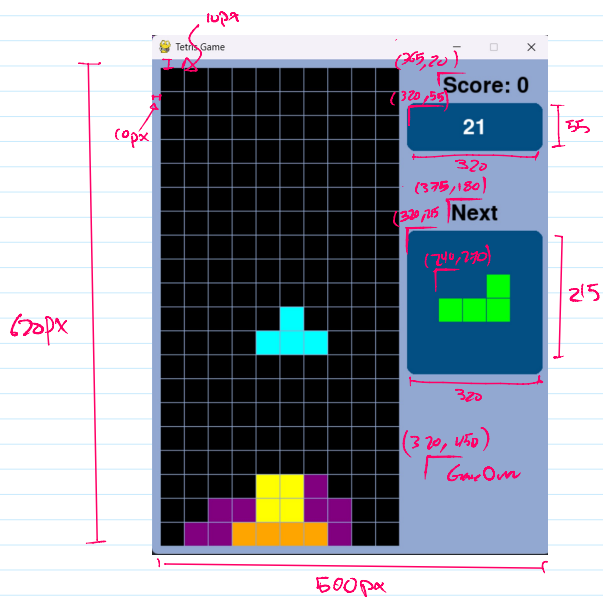


or



```
class UI:
    def __init__(self) -> None:
        self.title_font = pygame.font.Font(None, 40)
        self.score_surface = self.title_font.render(
            "Score: 0",
            True,
            Colors.BLACK.value,
        )
        self.next_surface = self.title_font.render(
            "Next",
            True,
            Colors.BLACK.value,
        )
        self.game_over_surface = self.title_font.render(
            "Game Over",
            True,
            Colors.BLACK.value,
        )
        self.score_rect = pygame.Rect(320, 55, 170, 60)
        self.next_rect = pygame.Rect(320, 215, 170, 180)

    def draw(self, screen):
        screen.fill(Colors.BACK_BLUE.value)
        screen.blit(self.score_surface, (365, 20, 50, 50))
        screen.blit(self.next_surface, (375, 180, 50, 50))
        if self.game.game_over:
            screen.blit(self.game_over_surface, (320, 450, 50, 50))
        pygame.draw.rect(screen, Colors.FRONT_BLUE.value, self.score_rect, 0, 10)
        pygame.draw.rect(screen, Colors.FRONT_BLUE.value, self.next_rect, 0, 10)

        score_value_surface = self.title_font.render(
            str(self.game.score),
            True,
            Colors.WHITE.value,
        )
        screen.blit(
            score_value_surface,
            score_value_surface.get_rect(center=self.score_rect.center),
        )
        self.game.next_tetromino.draw(screen, 240, 270) # draw next tetromino

    def update(self, game):
        self.game = game
```

Border 4, 11, 11  
 ↘ ↙ Border  
 ↘ ↙ Radius