Pisano Lucas BTS SIO 22

Rapport Projet Guess

Tout d'abord, j'ai mis en place ce qui était nécessaire à la réalisation du projet (PHP et composer). J'ai téléchargé le projet sur le site Github, fait la commande « composer Install » et « composer update ». Bien que j'aie rencontré quelques difficultés avec composer, j'ai fini par m'en sortir et enfin commencer le projet.

Une fois le projet prêt, j'ai tout d'abord testé la commande suivante : ./bin/phpunit pour voir si tout était opérationnel. Cette commande m'a affiché des erreurs ce qui est normal. Dans le fichier « CardTest » j'ai fait les « TODO » comme demandé.

tests > Core > CardTest.php

```php

    public function testColor()
    {
      $card = new Card('As', 'Trèfle');
      $this->assertEquals('Trèfle', $card->getColor());
      $card = new Card('As', 'Pique');
      $this->assertEquals('Pique', $card->getColor());
    }

    public function testCompareSameCard()
    {
      $card1 = new Card('As', 'Trèfle');
      $card2 = new Card('As', 'Trèfle');
      $this->assertEquals(0, CardGame32::compare($card1,$card2));
    }

    public function testCompareSameNameNoSameColor()
    {
      $card1 = new Card('As', 'Trèfle');
      $card2 = new Card('As', 'Coeur');
      $this->assertEquals(1, CardGame32::compare($card1,$card2));
    }

    public function testCompareNoSameCardNoSameColor()
    {
      $card1 = new Card('8', 'Trèfle');
      $card2 = new Card('As', 'Coeur');
      $this->assertEquals(-1, CardGame32::compare($card1,$card2));
    }

    public function testCompareNoSameCardSameColor()
    {
      $card1 = new Card('As', 'Trèfle');
      $card2 = new Card('8', 'Trèfle');
      $this->assertEquals(1, CardGame32::compare($card1,$card2));

    }
```

```
PS D:\sio22_dev\Projet Guess\guesswhat-master> php ./bin/phpunit tests/Core/CardTest.php
PHPUnit 7.5.20 by Sebastian Bergmann and contributors.

Testing App\Tests\Core\CardTest
........                                                            8 / 8 (100%)

Time: 291 ms, Memory: 6.00 MB

OK (8 tests, 10 assertions)
PS D:\sio22_dev\Projet Guess\guesswhat-master>
```

Une fois les tests complétés comme ci-dessus, je me suis attaqué à CardGame32.

En commençant par la fonction compare

```php
public static function compare(Card $c1, Card $c2) : int
{

  $c1Name = strtolower($c1->getName());
  $c2Name = strtolower($c2->getName());
  $c1Color = strtolower($c1->getColor());
  $c2Color = strtolower($c2->getColor());

  if ($c1Name === $c2Name ) {
    if($c2Color===$c1Color){


      return 0;
    }
    return (self::ORDER_COLORS[$c1Color]> self::ORDER_COLORS[$c2Color]) ? +1 : -1;
  }
  return (self::ORDER_NAMES[$c1Name]> self::ORDER_NAMES[$c2Name]) ? +1 : -1;
  return ($c1Name > $c2Name) ? +1 : -1;
}
```

Qui permet de comparer ne nom et la suite de deux cartes différentes. J'ai également ordonné les suites/couleurs et les valeurs/nom des cartes avec deux « const » (const order_color et const order_name).

Ensuite, la fonction factoryCardGame32, qui permet d'instancier un paquet de 32 cartes sous forme de tableau.

La fonction Shuffle permet de mélanger le jeu de 32 cartes précédemment créer. Et la fonction reOrder qui remet dans l'ordre le paquet.

```php
/**
 * Mélange le jeu de cartes
 */
public function shuffle(array $cardGame)
{
  shuffle($cardGame);
}

// remet le paquet dans l'ordre

public function reOrder(array $cardGame)
{
  sort($this->$cardGame);
}
```

Pour continuer, La fonction getCard permet de piocher une carte dans le jeu.

Pour tester ces fonctions, j'ai créer le fichier CardGame32Test.

Dans ce dernier, je teste la fonction testToString, TestToString2cards, TestGetCard, TestShuffle et TestreOrder avec phpunit.

```php
public function testToString()
{
  $cardGame = new CardGame32([new Card ('As','Trèfle')]);
  $this->assertEquals('CardGame32 : 1 carte(s)',$cardGame->__toString());
}

//teste la onction ToString mais avec 2 cartes
public function testToString2Cards()
{
  $cardGame = new CardGame32([new Card ('As','Pique'), new Card('Roi','Pique')]);
  $this->assertEquals('CardGame32 :  carte(s)',$cardGame->__toString());
}

//teste la fonction getCard qui pioche une carte dans le jeu
public function testGetCards(){
  $cardGame = CardGame32 ::factoryCardGame32();
  $index = 1;
  $card1 = $cardGame ->getCard($index);
  $this->assertEquals('Roi', $card1->getName());
  $this->assertEquals('Trèfle', $card1->getColor());
}

// teste la fonction shuffle (mélange)
public function testShuffle(){
  $cardGame = CardGame32::factoryCardGame32();
  $cardGame->shuffle();
  $cardGame2 = CardGame32::factoryCardGame32();
  $this->assertNotEquals($cardGame,$cardGame2);
}

//teste si la fonction reOrder fonctionne
public function testreOrder(){
  $cardGame = CardGame32::factoryCardGame32();
  $cardGame->shuffle();
  $cardGame->reOrder();
  $cardGame2 = CardGame32::factoryCardGame32();
  $this->assertNotEquals($cardGame,$cardGame2);
}
```

Cependant, lorsque je teste CardGame32Test avec phpunit