CheckErS2
Entrega final

Arthur Ferraz, Leonardo Schimpf, Vitor Araujo e Vítor Lourenço
Instituto de Computação - UFF
Junho de 2018

Agenda

- Testes
- Monitoramento e Controle
- Dificuldades e decisões tomadas
- Demonstração

Testes

Testes Unitários

- Testes realizados com o framework pytest
- Do modelo MVC, foram testados Model e Controller (para os unitários)
- Análise de cobertura

Mocks

Em alguns casos, abordagem com Mocks, em outros sem

```
def __init__(self, mode, args = None):
    """
        Class builder

        Parameters
        ----------
        mode : game mode (2 player or bot vs. human)
        args : bot level ('easy', 'normal', 'hard') or a file

        Returns
        -------
        A Rule class object
    """
    self.board = Board()
    if mode == 'file':
        self._load_file(args)
    else:
        self.players, self.turn_player = self._set_players(mode, args)
        self._init_board(self.board, self.players)
```

```
@patch('src.mvc.Controller.rules.Rule._load_file')
def test_rules_init(self, load_file_mock):
    args = 'teste'
    rules_with_load = Rule('file', args)
    assert load_file_mock.call_count == 1
    assert load_file_mock.called_with(args)
    rules_wo_load = Rule('human')
    assert load_file_mock.call_count == 1
```

Sem mocks

```python
def get_qty_draughts(self):
    qty = 0
    for piece in self.pieces:
        if piece.is_draughts:
            qty += 1
    return qty
```

```python
def test_player_get_qty_draughts(self):
    player = Player("p1", "red", 1)
    assert player.get_qty_draughts() == 0
    pieces = [Piece("p1") for i in range(5)]
    player.set_pieces(pieces)
    assert player.get_qty_draughts() == 0
    player.pieces[0].turn_draughts()
    assert player.get_qty_draughts() == 1
```

Cobertura de testes

- Modelo: 99% de cobertura
- Controller: 35% de cobertura

Exemplo de cobertura

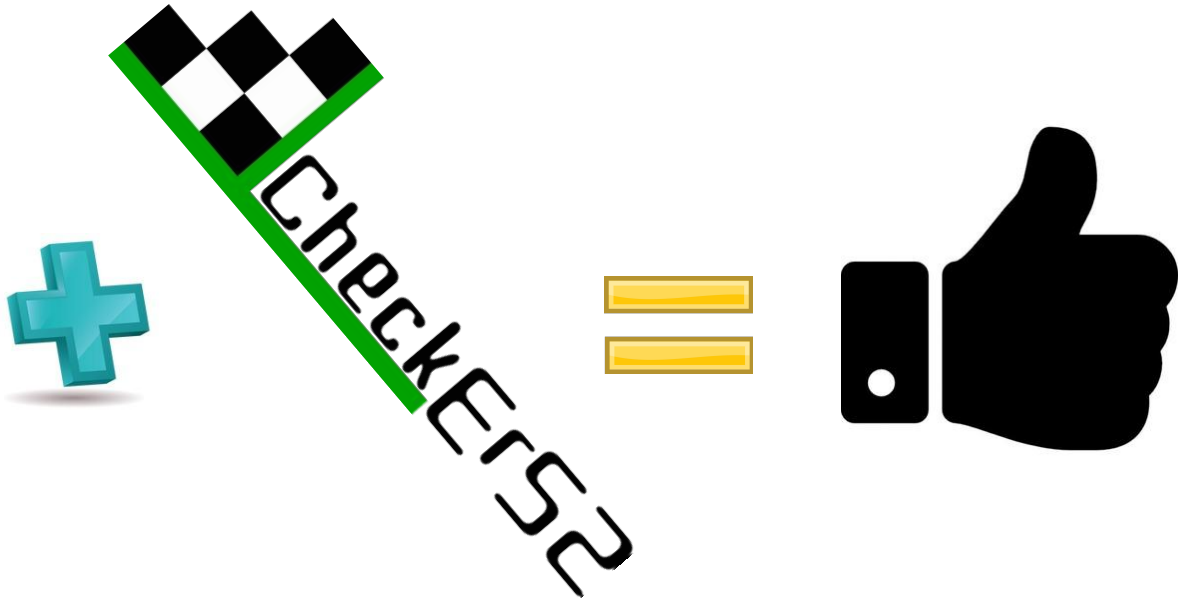
```
leosch@leonote:~/Desktop/uff/es2/CheckErS2/tests$ ./rodaTeste.sh
============================= test session starts ==============================
platform linux -- Python 3.5.2, pytest-3.6.1, py-1.5.3, pluggy-0.6.0
rootdir: /home/leosch/Desktop/uff/es2/CheckErS2/tests, inifile:
plugins: cov-2.5.1
collected 24 items

test_model.py ........................                                  [100%]

---------- coverage: platform linux, python 3.5.2-final-0 -----------
Name                                                                                 Stmts   Miss  Cover   Missing
------------------------------------------------------------------------------------------------------------------
/home/leosch/Desktop/uff/es2/CheckErS2/src/mvc/Model/__init__.py                         0      0   100%
/home/leosch/Desktop/uff/es2/CheckErS2/src/mvc/Model/board.py                           19      0   100%
/home/leosch/Desktop/uff/es2/CheckErS2/src/mvc/Model/bot.py                             95      2    98%   53, 101
/home/leosch/Desktop/uff/es2/CheckErS2/src/mvc/Model/movement.py                        40      0   100%
/home/leosch/Desktop/uff/es2/CheckErS2/src/mvc/Model/piece.py                           12      0   100%
/home/leosch/Desktop/uff/es2/CheckErS2/src/mvc/Model/player.py                          18      0   100%
------------------------------------------------------------------------------------------------------------------
TOTAL                                                                                  184      2    99%


========================== 24 passed in 0.32 seconds ===========================
```

Testes de Aceitação



Teste de Usabilidade

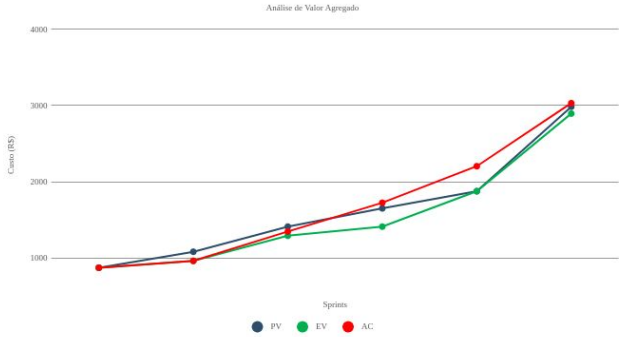
- Atividade 1
  - Objetivo: Verificar o fluxo principal
  - Enunciado: Você deve movimentar o jogo até comer uma peça
- Atividade 2
  - Objetivo: Verificar seleção de peça
  - Enunciado: Você deve selecionar uma peça e, depois, selecionar outra
- Atividade 3
  - Objetivo: Verificar a eliminação de múltiplas peças
  - Enunciado: Você deve interagir em uma eliminação múltiplas peças
- Atividade 4
  - Objetivo: Verificar o fim de jogo
  - Enunciado: Você deve jogar até o fim do jogo

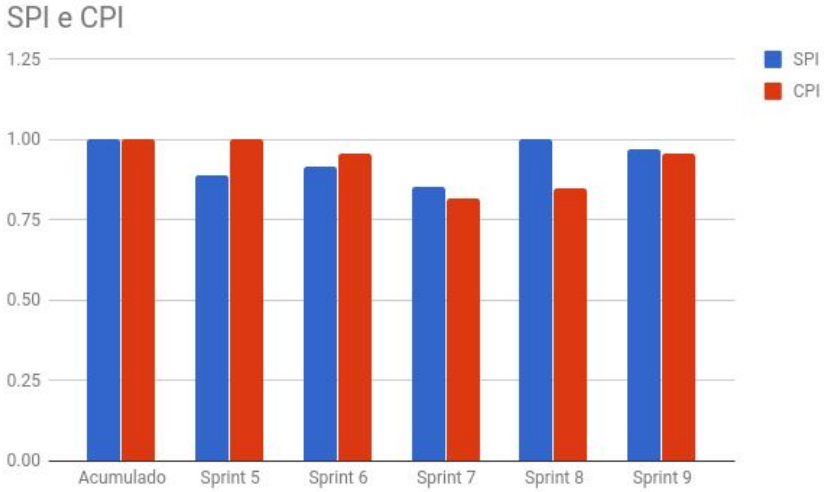
Teste de Usabilidade

- Resumo:
  - De forma geral, os usuários entrevistados executaram o fluxo principal de forma natural e sem precisar de auxílio;
  - Metade dos usuários entrevistados tiveram dificuldades para selecionar uma segunda peça. O movimento natural observado foi de tentar selecionar uma segunda peça ao invés de deselecionar a atual e depois selecionar a segunda peça;
  - Os usuário entrevistados elogiaram a remoção das peças apenas ao fim da interação;
  - Todos usuários criticaram a ausência de uma mensagem *user-friendly* de fim de jogo.

Monitoramento e Controle

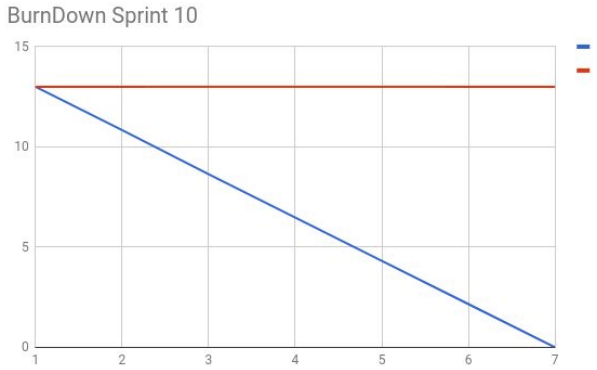
Monitoramento e Controle - Sprint 9

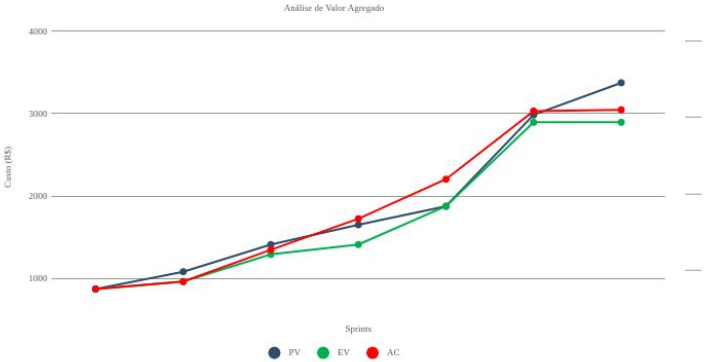


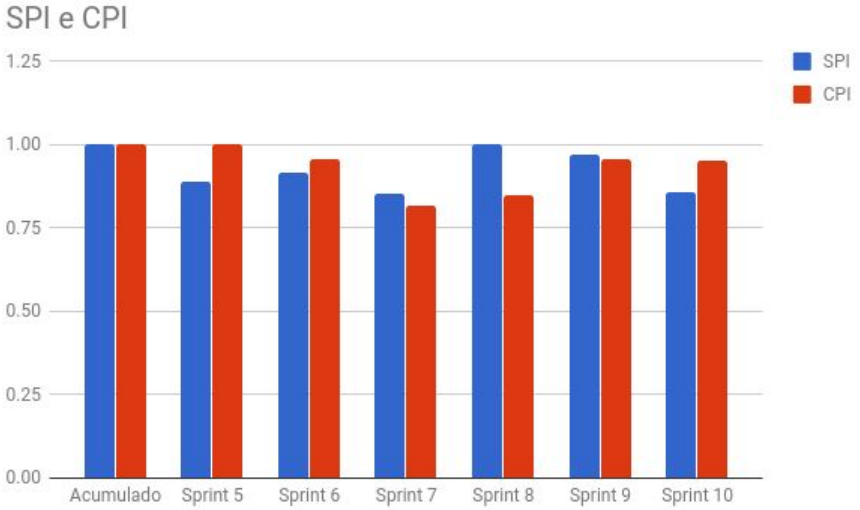


- Criação da IA
- Implementação da condição de empate

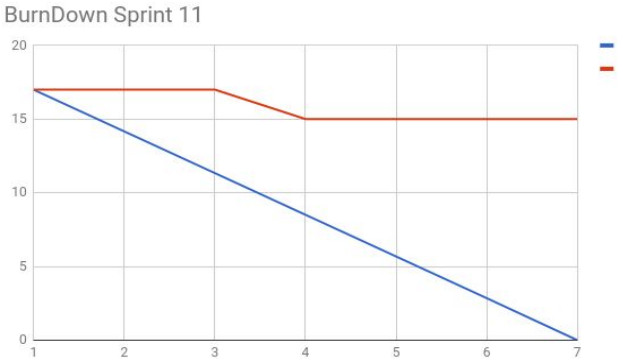
Monitoramento e Controle - Sprint 10

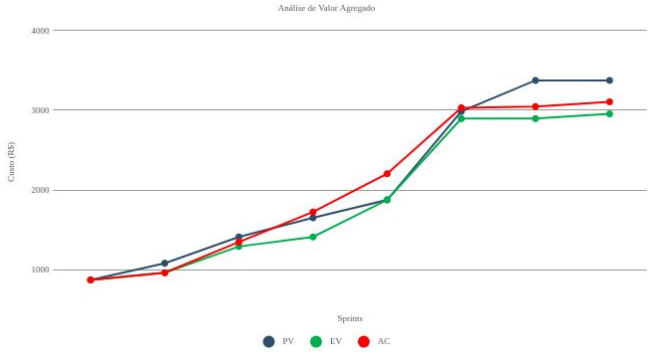


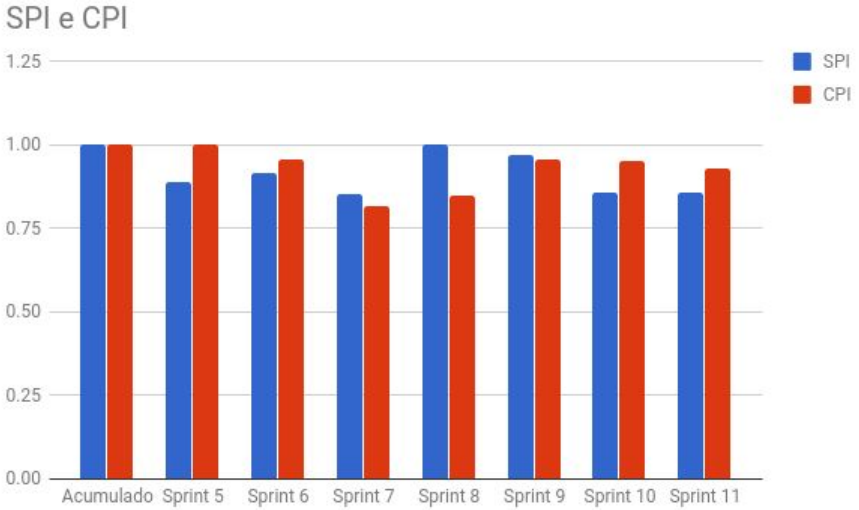




Monitoramento e Controle - Sprint 11

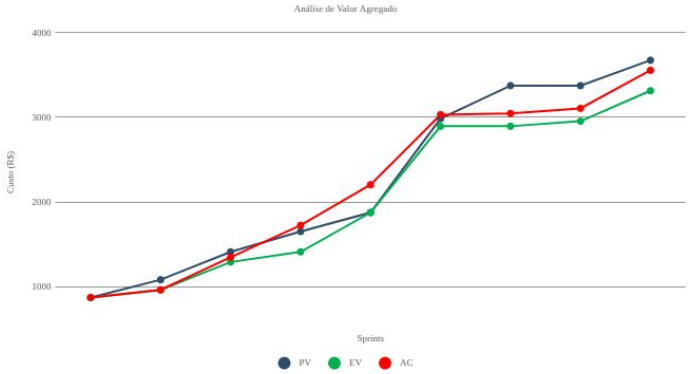


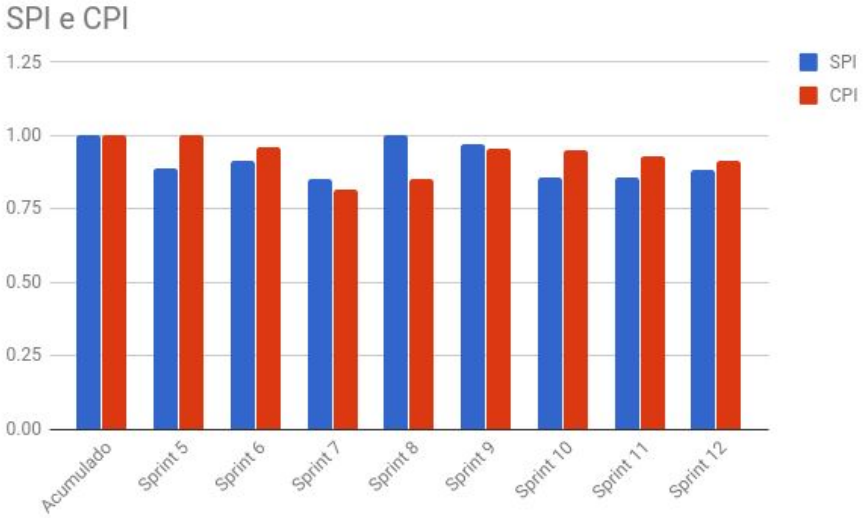




- Início de testes e correção de bugs
- Menu do jogo

Monitoramento e Controle - Sprint 12





- Testes unitários
- Teste de usabilidade
- Definição de hiperparâmetros da IA

Dificuldades e decisões tomadas

Dificuldades e decisões tomadas

- Integração em diferentes ambientes
  - Abandono de JS e desenvolvimento completo em Python
- Dificuldades para mensurar custos
  - Custo de testes mal mensurado e interferência direta no CPI final
- Horários e comunicação interna
  - Horários heterogêneos e irregulares e comunicação feita principalmente de forma virtual
- Integração e entrega contínua
  - Dificuldade para configuração do Travis CI e impossibilidade de utilizá-lo

Demonstração


CheckErS2