PyCards

Project Team:	Aravi Premachandran, Michael Lee and Nikhil Patel
Course:	SFWR ENG 3XA3
Date:	November 27, 2016

Introduction

- Objective: redevelop an interesting open-source project using a rational design process
- Motivation: create an engaging source of entertainment in the form of a collection of solitaire card games
- Stakeholders: client, end-users, fellow software developers and contributors

The Zen of Python

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
*Although never is often better than *right* now.*
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!

Design

- Model-View-Controller design pattern
- Customizable interface
- 100% Python implementation

Demonstration

Discussion

- Challenges
  - GUI implementation
  - Principle of least privilege
- Notable Qualities
  - Portability
  - Modularity

Portability

- Cross-platform
  - OSX, Windows 10, Ubuntu
- No installation required
  - Distributable as a single executable file

Maintainability

- Modular design
- Design for change
- Automated build and testing
  - py2exe and unittest.py

Questions?

- Feel free to ask us questions until the end of our allocated time slot

Citations

- *https://www.python.org/dev/peps/pep-0020/*