# Heuristic Evaluation

## Heuristics (Nielson: <http://www.nngroup.com/articles/ten-usability-heuristics/>)

1. **Visibility of system status**
   * The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
2. **Match between system and the real world**
   * The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
3. **User control and freedom**
   * Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
4. **Consistency and standards**
   * Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
5. **Error prevention**
   * Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.
6. **Recognition rather than recall**
   * Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
7. **Flexibility and efficiency of use**
   * Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
8. **Aesthetic and minimalist design**
   * Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
9. **Help users recognize, diagnose, and recover from errors**
   * Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
10. **Help and documentation**
    * Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

# Heuristic Evaluation

## Severity Ratings (Nielson: http://www.nngroup.com/articles/how-to-rate-the-severity-of-usability-problems/)

The severity of a usability problem is a combination of three factors:

* The **frequency**with which the problem occurs: Is it common or rare?
* The **impact**of the problem if it occurs: Will it be easy or difficult for the users to overcome?
* The **persistence**of the problem: Is it a one-time problem that users can overcome once they know about it or will users repeatedly be bothered by the problem?

The following 0 to 4 rating scale can be used to rate the severity of usability problems:

0 = **I don't agree** that this is a usability problem at all

1 = **Cosmetic problem only**: need not be fixed unless extra time is available on project

2 = **Minor usability problem**: fixing this should be given low priority

3 = **Major usability problem**: important to fix, so should be given high priority

4 = **Usability catastrophe**: imperative to fix this before product can be released

# Heuristic Evaluation

Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Project Title:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Interface** | **Issue** | **Heuristic(s)** | **F**requency 0 (rare) to 4 (common) | **I**mpact 0 (easy) to difficult (4) | **P**ersistence 0 (once) to 4 (repeated) | **Severity** = Sum Total of F+I+P /3 |
| Gameplay | Pac-Man does not interact with the ghosts | Consistency and standards | 4 | 4 | 4 | 4 |
| Gameplay | No coins for Pac-Man to eat | Consistency and standards | 4 | 4 | 4 | 4 |
| Gameplay | The limited visibility twist has not been implemented yet | Aesthetic and minimalist design + consistency and standards | 3 | 3 | 3 | 3 |
| Gameplay | Pac-Man’s spawn location should not be inside a wall | Error prevention + Aesthetic and minimalist design + Consistency and standards | 1 | 2 | 2 | 2 |
| Gameplay | There should be a theme to explain the gameplay twists | Aesthetic and minimalist design | 2 | 1 | 3 | 2 |
| Gameplay | There is no music throughout the gameplay | Aesthetic and minimalist design | 4 | 1 | 1 | 3 |
| Gameplay | It could be more fun to have full visibility before having limited visibility to make the game more difficult as it progresses | Aesthetic and minimalist design | 2 | 2 | 3 | 2 |
| Menu | There is no start menu | Help and documentation + Visibility of system status | 3 | 3 | 2 | 3 |

# Heuristic Evaluation

Name:\_\_\_\_TA’s evaluation\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Project Title:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Interface** | **Issue** | **Heuristic(s)** | **F**requency 0 (rare) to 4 (common) | **I**mpact 0 (easy) to difficult (4) | **P**ersistence 0 (once) to 4 (repeated) | **Severity** = Sum Total of F+I+P /3 |
|  | The ghosts do not interact with Pac-Man. | Consistency and standards | 4 | 4 | 4 | 4 |
|  | There are no coins. | Consistency and standards | 4 | 4 | 4 | 4 |
|  | The theme could make more sense. Why are there glowing multi-coloured walls? | Aesthetic and minimalist design | 1 | 1 | 4 | 2 |
|  | The controls feel slightly laggy/delayed. There is a buffered input. I do not like how the keypresses are saved (e.g. when down is pressed before Pac-Man can move down, the keypress is saved until Pac-Man can move down). The movement could be snappier. I would prefer it to handle like the game Snake. Maybe the game could have the option to change how the controls work. I would like it to be easier to override the saved keypresses. Maybe this game could benefit from a ‘coyote time’ where there is a small window to change direction before fully committing to a new direction. Maybe Pac-Man could have a buffered input for only a few milliseconds. | User control and freedom | 3 | 2 | 2 | 2 |
|  | The game could be more interesting with teleportation portals in the map. | aesthetic and minimalist design | 2 | 2 | 0 | 1 |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |