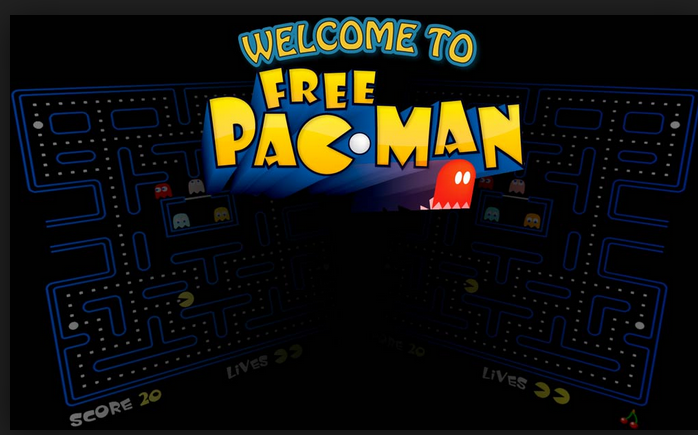
Software Requirements Specification:

For **Pac**-**Man** 

Version 1.0

     Version 1.0

Prepared by Spec Team

David Chan

Jinglin Tan

Chaudhary Riasat

Anva Kondakciu

                                                                                                                                              md raju ahamed

**GENERAL OVERVIEW**

**Development Tools**

* HTML5
* CSS3
* JavaScript E6

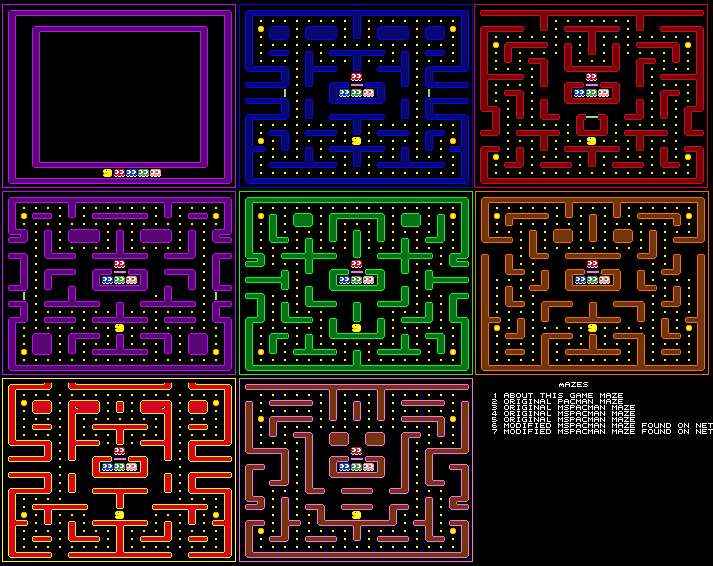
**Interface**

compatible with all popular browsers. Ex. Firefox, Chrome, Internet Explorer.

**Game Description**

The player navigates Pac-Man through a maze containing dots, known as Pac-Dots, and four multi-colored ghosts: The goal of the game is to accumulate as many points as possible by collecting the dots and eating ghosts. When all of the dots in a stage are eaten, that stage is completed and the player will advance to the next one.

examples of possible type of layout design that can be implemented for graphic:



**Required Elements**

There should be distinctive types of characters: Pac-Man (yellow), Ghosts (pink, baby blue, purple, and red. (possible can be trustees goblins, image below for graphic ), Gobbling fruits.

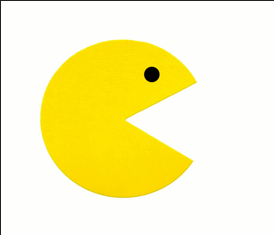
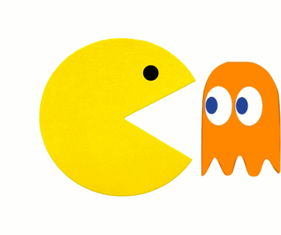
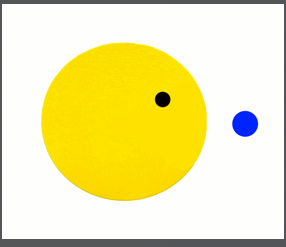
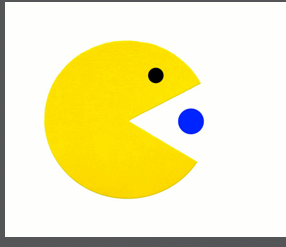
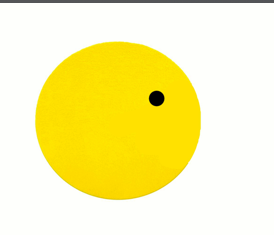
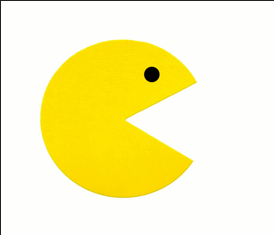
adding power pellets or push button to switch into power mode that will have a

time to active and deactive

to consider thinking in terms of design ease to have a button to toggle on and off when you can eat the ghosts with a time delay between when you can active and reactive again. (sounds effect of eating the pellets and ghosts )

trustee to have screaming effect after eaten. (graphic and sounds)

include images of pac-man in different images representing eating and moving options below to demonstrate some preliminary image concepts: will need an Pac-man image for up, right, down, and left.



PACMAN

* Position on board (x,y)
* Direction (w,a,s,d) and (up arrow, left arrow, right arrow, down arrow)
* Speed // possible two mode easy and hard.

**Scoring**

Achieving highest score or to complete a stage. include an image below for high score.

add a high score chart below.



Ghost (Trustees) +100 points

image of trustee below ghost to be eaten in the game with additional points.



Food objective

pellets +10 points and image below and controller sequence

bool isFood (x,y)

SAMPLE GAME INSTANCE

Pacxman { row x, column y};

Ghost [ { row x, column y};

{ row x, column y};

{ row x, column y};

]

**Revision History**

Name Date Reason For Changes

David Chan 2/19/19 started the layout for the spec requirement.

Some details of characters, game logic and

Concepts. 1.1,1.2,1.3,1.5

David Chan 2/20/19 communicated with graphic and backbone

implement some design and game concepts

Jinglin Tan 2/21/19 added section 2.1 - 2.7

Jinglin Tan 2/25/19 revised section 2.5, 2.6, 2.7

Chaudhary Riasat 2/30/19 add section 3.1,3.2

md raju ahamed 3/1/2019 communicated with graphic and backbone

to decide on the game play modes.

Anva Kondakciu 3/4/2019 Completed remaining parts of section 3 and

all of section 4. Game features and interface

revised

David Chan add addition scope to the overview of the

Anva Kondakciu 3/5/2019 game food and scoring.

Jinglin Tan 3/6/2019 add controller and movement updates to

section 2.6 and after speaking with graphics

they would like to add laser beam effect and

updated over on additional mode on game

speed

Anva Kondakciu 3/8/2019 communicated with graphic and updated

overview of the game play.

updated overview on game theory and

movements on the overview section.

Chaudhary Riasat

& md raju ahamed 3/9/2019 add in the overview the game instance into

game overview

David Chan 3/19/2019 making changes to the overview.

Jinglin Tan 3/25/2019 add addition control unit and color

Anva Kondakciu 4/1/2019 add browser interface to overview

David Chan 4/5/2019 add color spec and add additional

control units to section 1.3 and overview

Chaudhary Riasat &

md raju ahamed 4/8/19 created new append section.

adjust the color spec

**Table of Contents**

**Table of Contents**

**Revision History**

**1. Introduction**

1.1 Purpose

1.2 Document Conventions

1.3 Intended Audience and Reading Suggestions

1.4 Product Scope

1.5 References

**2. Overall Description**

2.1 Product Perspective

2.2 Product Functions

2.3 User Classes and Characteristics

2.4 Operating Environment

2.5 Design and Implementation Constraints

2.6 User Documentation

2.7 Assumptions and Dependencies

**3. External Interface Requirements**

3.1 User Interfaces

3.2 Hardware Interfaces

3.3 Software Interfaces

3.4 Communications Interfaces

**4. System Features**

**5. Append**

**1. Introduction**

**1.1 Purpose**

* Achieving highest score or to complete a stage.
* Version 1.0

**1.2 Document Conventions**

* Followed Software Requirements Specification for <Project> by IEEE Version 1.0

**1.3 Intended Audience and Reading Suggestion**

* This documentation is intended for end users, developers and project managers.  
  All types of reader who want to learn the scope of the product and the detailed implementation of the software are suggested to read sequentially each chapter.
* **End users is recommended to read Append for updates and changes.**

**1.4 Product Scope**

* to develop and recreate a popular Pac-Man game for a class project where Pac-Man is chasing eating pellets and trustees to achieve a high scores.
* Game end when pac-man get eaten by the (ghost trustees)
* object of the game is to get the highest score before losing all the lives in the game.

**1.5 References**

* [**https://en.wikipedia.org/wiki/Pac-Man**](https://en.wikipedia.org/wiki/Pac-Man)
* **Pac-Man**​ is arcade game designed by Toru Iwatani of Namco and first released in Japan as ***Pac-Man*** in May 1980.

**2. Overall Description**

2.1 **Product Perspective**

the main game screen consist of start mode button

options of difficulty to select speed for slow or fast (easy and hard).

main game screen will have walls to have paths for Pac-Man to direct his movement directions and possible consider an wormhole entrance that will teleport Pac-Man into another section of the of the maze

the objective of the game is to achieve highest score before dying(eaten by trustee).

2.2 **Product Functions**

the user need to be able move up, down, left, right to eat pellets and avoid ghosts.

2.3 **User Classes and Characteristics**

the end user is the class which will play and test the game in development phases

we expect the user to have some basic understand of the popular game Pac-Man and

understand to make changes to possible add additional twist the already popular game.

2.4 **Operating Environment**

the game should be operated in a web browser or desktop computer the computer be equipped with functioning audio and visual output and a working mouse we anticipate a minimum 1024 x 768 screen resolution and 24-bit color depth on the

computer

2.5 **Design and Implementation Constraints**

minimum must include the following:

HTML5

CSS3

JavaScript E6

compatible with popular browsers. Ex. Firefox, Chrome, Internet Explorer.

2.6 **User Documentation**

the game will start click of the mouse on the start game button

possible option of different modes

example of a different mode would be pac-man speed. To have two different levels.

PACMAN

* Position on board (x, y **coordinates)**
* Direction (w,a,s,d)
* Direction (w,a,s,d) and (up arrow, left arrow, right arrow, down arrow)

**CONTROLLER**  
 void initGame ()

void initInterface ()

startEventListener ()

2.7 Assumptions and Dependencies

graphic would like to implement with additional time to allow Pac-man to power up and

shoot laser beams at the ghost.

**3. External Interface Requirements**

**3.1 User Interfaces**

* A graphic User interface is needed for the Introductory, Instruction, Game, and the Game Over Screen.
* The Introduction screen must show the title of the game and a button to get to the next page
* the instruction screen must display the score

the game, pac-man will a have more than 1 lives.

* the game over screen must display the total amount of points accumulated after death.

**3.2 Hardware Interfaces**

* this game must be compatible with window 7, & 10, and Mac OS
* requires mouse and keyboard
* CPU: 1.00 GHZ
* Memory (RAM): 2.00 GB
* System Type: 64 bit operating system
* Graphic card: minimum intel or amd cpu graphic chip
* Hard Drive: 1 GB Available space.

**3.3 software interfaces**

this game must be able to keep a highest score and store the data on a server. possible a database implementation may need to be require like SQL database to save the data on a local create database or server.

**3.4 Communications Interfaces**

The control for the communication element is managed through the web application program. Javascript libraries and other modules are used to serve this purpose.

**4. System Features**

**4.1 System Feature 1**

To complete each level of the game, the player must use all four arrows to navigate the maze. Ghosts and point options increase in more complex episodes when areas of the maze are unlocked successfully.

**4.2 System Feature 2 (and so on)**

The game emulates a later version of Pac-Man known as Pac-Man: Adventures in Time. Pac-Man will travel through different time periods in correct order of time. Pac-Man is on a mission in this game to go back to his current time period before the ghosts can reach him.

**5. Append**

Graphic 4/2/19

**Large Pellet:**

**Dimensions: 115x170**

**Color Depth: 32 bit**

**Color Range: gold and green**

**Border Outline: black**

**Image:**

****

**Small Pellet:**

**Dimensions: 127x127**

**Color Depth: 32 bit**

**Color Range: gold and green**

**Border Outline: black**

**Image:**

****

**Blood Splatter:**

**Dimensions: 150x127**

**Color Depth: 32 bit**

**Color Range: red**

**Border Outline:none**

**Image:**

****

**Trustee Ghosts:**

**Dimensions: 175x101**

**Color Depth: 24 bit**

**Color Range: black, orange, turquoise, yellow, pink, peach, white, light brown, tan, grey.**

**Border Outline:black**

**Images:**

****

**Regular Pacman:**

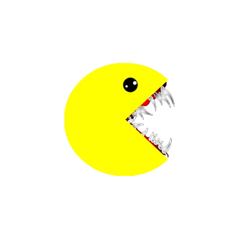
**Dimensions: 125x125**

**Color Depth: 24 bit**

**Color Range: yellow, red, white, grey.**

**border outline: none**

**Image:**

****