

REQUIREMENTS PAC-MAN

Group 29

CONTENTS

1. Functional Requirements	2
1.1. Must Haves	2
1.2. Should Haves	2
1.3. Could Haves	2
1.4. Would/Won't Haves	3
2. Non-Functional Requirements	4

1. Functional Requirements

1.1. Must Haves

- The player shall be able to register with a username and password.
- The player shall be able to login with a username and password.
- The player shall be able to start a new game of pacman
- The game shall be able to retrieve the player statistics with the correct username and password combination.
- The game shall be able to move NPC¹ units with predefined rules in all four directions.
- The game shall have a controllable player unit.
- The player shall be able to move pac-man with key presses in all four directions with the WASD keys.
- The game shall have a board consisting of a grid with variable size.
- The game shall be able to detect the important collisions between units.
 - Player-pellet: increment points by 10, delete pellet from board.
 - Player/Ghost-wall: restrict movement of unit towards cell with wall.
 - Player-Ghost: kill player, stop game.
 - Ghost-pellet: nothing happens, ghosts should ignore pellets and leave them as is.
- The game shall end when a player and NPC unit collide (level lost).
- The game shall end the level when all pellets are collected (level won).

1.2. Should Haves

- The game shall be able to keep track of the amount of consumed pellets.
- The game shall be able to parse a string with predefined characters into a map.
- The player shall be able to start a new game with predefined unit positions.
- The game design shall have a wraparound.
- The game shall be able to detect the additional collisions between units.
 - Player-fruit: increment points by 100, 300 or 500 if the fruit is a cherry, strawberry or apple respectively. delete fruit from board.
 - Player-power pellet: increment points, delete pellet from board, grant power up status to player.
 - Powered up player-ghost: increment points by 200, 400, 800, or 1600 if it's the first, second, third, or fourth ghost eaten during a single power-up status respectively. Kill ghost.

¹ NPC stands for Non-Player Character. In this case, we are referring to the ghosts.

- Ghost-power pellet: nothing happens, ghosts should ignore power pellets and leave them as is.
- The game shall have the top 10 scores of all players in some database.

1.3. Could Haves

- The player shall be able to choose between using arrow keys and WASD keys to move Pac-Man.
- The game shall consist of multiple levels with units placed upon it.
- The game shall have pausing functionality.
- The levels shall have fruits that provide extra points alongside normal pellets.
- The game shall have power pellets that provide a power up status² alongside normal pellets.
- The player shall have a temporary power up status after consuming power pellet.
- The ghosts shall respawn in their base after being killed and they shall be able to get out after the power-up status of Pac-man wears down.
- The game shall have super pellets that provide a super status³ alongside normal pellets.
- The player shall have a temporary power up status after consuming super pellet.
- The main menu shall display a personalized welcome message for the user after logging in.
- The game shall have an inverted controls mode.
- The game shall have Survival mode (increasing ghosts as you advance through levels).
- The player shall have the option to select Difficulty (easy, normal, hard).

1.4. Would/Won't Haves

- The game shall have online matchmaking.
- The game shall have teleportation.
- The game shall have a mobile version.
- The game shall have Multiplayer mode (on same screen with one person controlling NPC).

² During the power up status, Pac-man is invincible and ghosts can be eaten for points.

³ During the super status, Pac-man can eat walls to run away from ghosts.

2. Non-Functional Requirements

- The game shall be implemented in Java 12.
- A working prototype of the game shall be delivered on December 6 2019.
- The Scrum methodology shall be applied.
- The code shall have at least 75% of meaningful branch test coverage.
- Checkstyle within CI/CD pipelines shall be used for the code.
- PMD within CI/CD pipelines shall be used for the code.
- Metrics report within CI/CD pipelines shall be used for the code.
- Static analysis within CI/CD pipelines shall be used for the code.