



SINISTER TRANSISTOR

BY GREG KELSO, MARK BOUTWELL, JOEL GARDYASZ

SINISTER TRANSISTOR

- What is it?
- 2D game shoot 'em up
- Player is a transistor
- Fight off resistors and capacitors
- Powerups available

SINISTER TRANSISTOR

- Why?
- We were interested in game design
- Wanted to put our own take on a game

SINISTER TRANSISTOR

- There are many similar games out there
- But none with an evil transistor
- Fills the need of people wanting games where they can fight against resistors

FUNCTIONAL REQUIREMENTS

- The player shall be able to move up, down, left, and right, unless stopped by a wall.
- The software shall feature music and sound effects.
- The player shall be able to navigate into other areas and rooms in their environment.
These rooms shall load and open once the player has indicated their desire to enter.
- The player and enemies shall have a set amount of health. If this is depleted, the player/enemy shall die. If the player dies the game shall end.

INTERFACE REQUIREMENTS

- The user shall be able to start a new game.
- The player shall be able to pause the current game to pull up a pause menu.

PHYSICAL ENVIRONMENT REQUIREMENTS

- The application shall run on a PC using the latest version of either Windows, OSX, or Linux
- The application should support keyboard and mouse input

USER AND HUMAN FACTORS REQUIREMENTS

- The game shall support users of all skill levels and of all ages.
- The game shall accommodate all players by having an ability to increase or decrease audio and font size.
- The software should be bug free. The game should not let users exploit bugs to cheat and achieve a higher score.

DOCUMENTATION REQUIREMENTS

- The player shall be able view a manual consisting of game features and mechanics.
- All game documentation shall be put online for the users to explore all aspects of the game and its design and implementation.

DATA REQUIREMENTS

- Entity Controllers shall store all relevant data to the object it is attached to.
- Unity built-in functions shall be used when relevant.

RESOURCE REQUIREMENTS

- The system shall require no more than three software engineers to complete the project.
- The project shall be completed with Unity2D, coded in C#, and managed with Git version control.
- The project, along with its required documentation and presentation brief, shall be completed no later than 22 April 2016.
- The project shall be completed with only free software engineering tools.
- The overall size of the project, including necessary documentation, shall take no more than 1GB of memory.

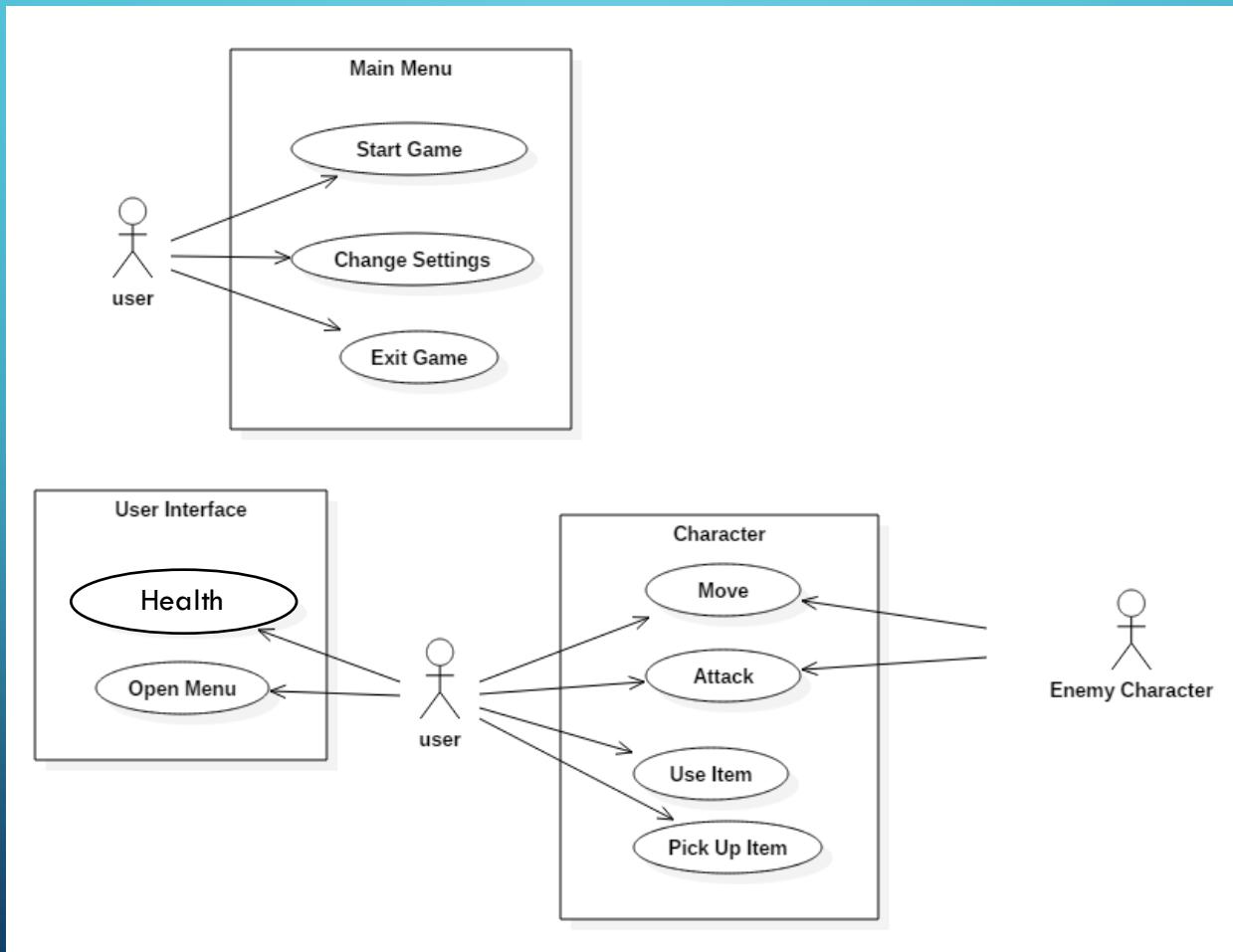
SECURITY REQUIREMENTS

- The system shall remain secure during development.
- The system shall be stored and version controlled using Git.
- The system shall execute as a standalone application for Windows, OS X, and Linux devices.

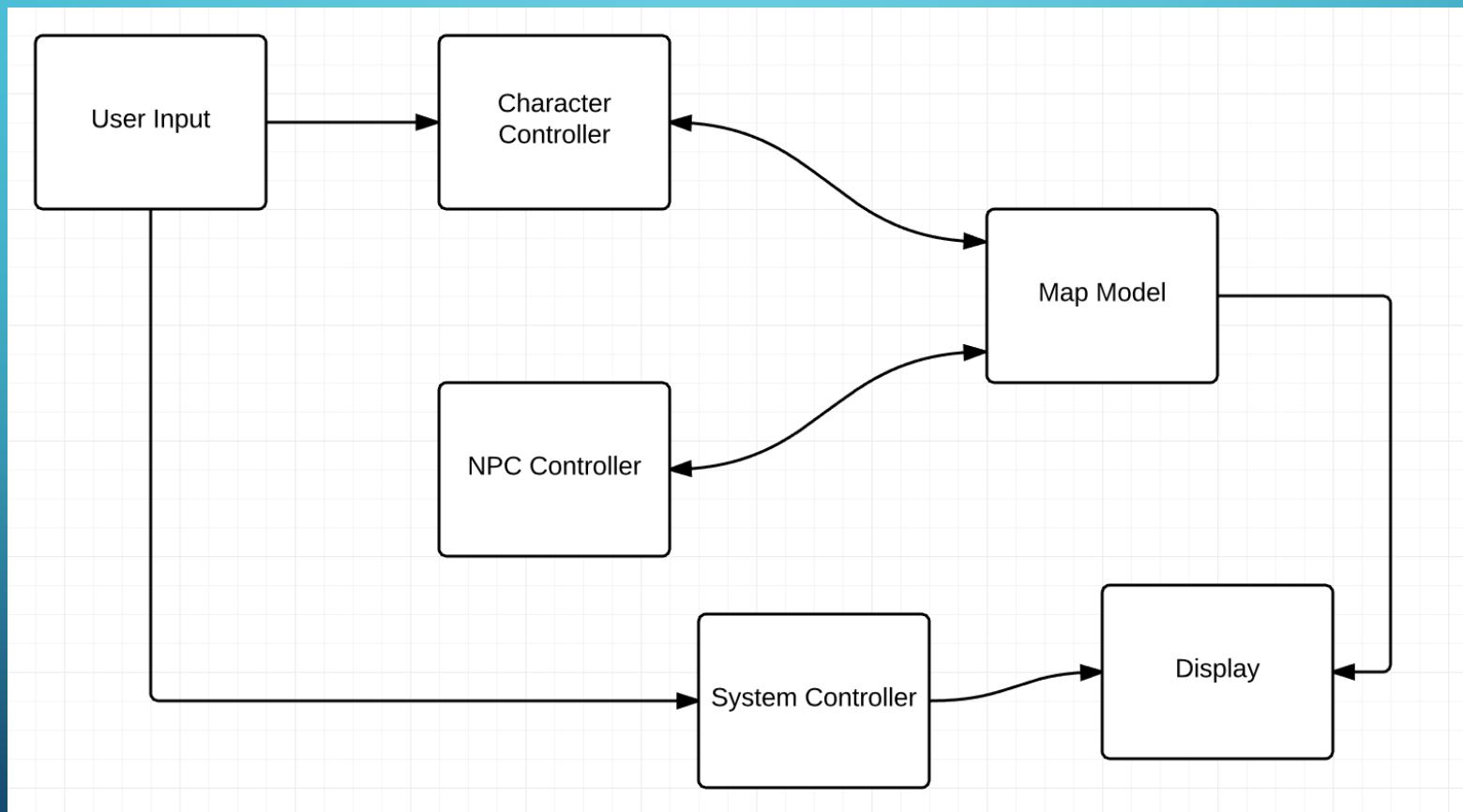
QUALITY ASSURANCE REQUIREMENTS

- The system shall be readable to all speakers of the English language.
- The system shall minimize the required time to test.
- The user shall meet the minimum Unity System Requirements in order to play the video game.

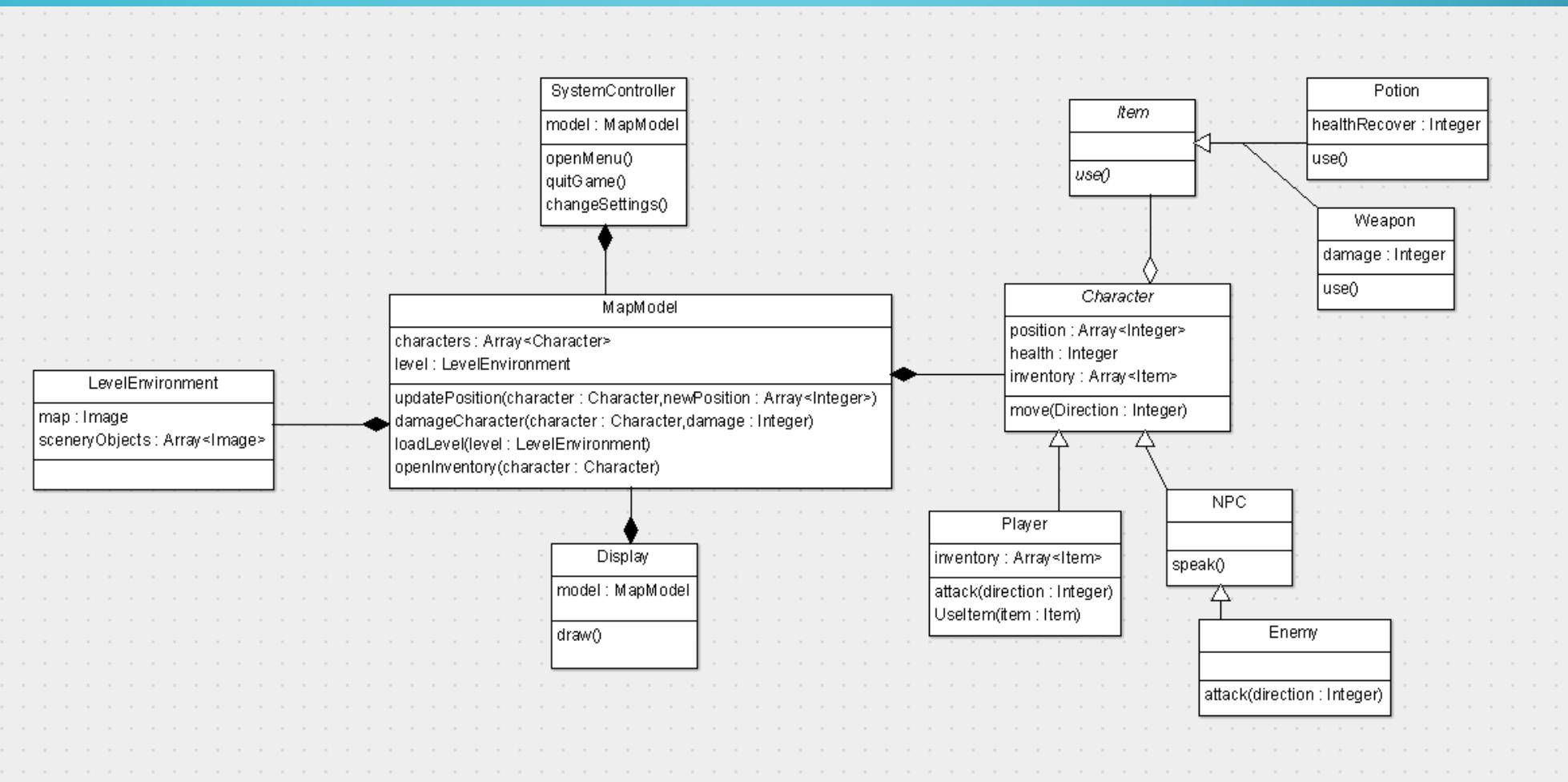
USE CASE DIAGRAM



HIGH LEVEL ARCHITECTURE



CLASS DIAGRAM



SACRIFICED REQUIREMENTS

- The game shall accommodate all players by having an ability to increase or decrease audio and font size.