Ryan Maloney, Alex Zilbersher (Team Members)

Dr. Ken Baclawski

CS 3520

September 26, 2016

Topic and Use Case Description of Project

**System:** Simple Platforming Game

* **Name:** Simple Platforming Game
* **Authors:** Ryan Maloney, Alex Zilbersher
* **Description:** This system is a platforming game in which the user can control a character which moves throughout the game world. The system includes the mechanics involved in moving the character and detecting with what the character does or does not collide with. Starting the game is not included within the use cases, however if a player chooses to restart the game from within the system after losing is included as it occurs as part of the system. Objects within the game with which the player character can interact will be referred to as items or objects.
* **Organization:** Northeastern University (CS 3520)
* **Creation Date:** September 26, 2016

**System Actors**

* **Player**
  + **Description:** The player is the only human and/or system outside of the game which can interact with the system. The player controls the main character and makes decisions based off of the inputs they use while playing the game.

**Use Cases**

* **Character Movement**
  + **Description:** The way the player moves the character in the game.
  + **Step-By-Step Description:**
    - **Includes** Jump Character
    - The player inputs a command to move the character.
    - The system recognizes that command.
    - The system moves the character.
* **Jump Character**
  + **Description:** Allows the character to jump in the air as an alternative form of movement.
  + **Step-By-Step Description:**
    - The player inputs a command to make the character jump.
    - The system reads and recognizes that command.
    - The system makes the character jump.
* **Avoid Damage**
  + **Description:** The character avoids taking damage by not colliding with any damaging obstacles.
  + **Step-By-Step Description:**
    - **Extends** Jump Character and Move Character.
    - The system checks if the character is touching an obstacle.
    - If the character is not, then the character’s state is not changed and damage is avoided.
* **Avoid Gaps**
  + **Description:** The character avoids falling into automatically fatal gaps/avoids missing any platforms.
  + **Step-By-Step Description:**
    - **Extends** Move Character and Jump Character
    - The system checks that the character has not fallen under a certain elevation.
    - If the player has not, then the player state is not changed and death is avoided.
* **Take Damage**
  + **Description:** The character within the game takes damage when it comes in contact with an object/obstacle which is recognized as hostile or damaging by the system.
  + **Step-By-Step Description:**
    - The system checks if the character is touching any damaging objects.
    - If the character is, then its health is lowered until the character moves to a new level or picks up a health boost.
* **Player Death**
  + **Description:** The character dies due to either not avoiding gaps or by hitting too many damaging obstacles and therefore its health being lowered to zero.
  + **Step-By-Step Description:**
    - **Includes** Losing a Life
    - The system checks if the character’s health is above zero.
    - If it registers as zero or below zero, the character dies at its current location within the game/system.
* **Score Decrease**
  + **Description:** The player’s score is decreased when the character takes damage or dies.
  + **Step-By-Step Description:**
    - The system checks if the character has died or taken damage.
    - If the character has, then the player’s score is lowered depending on the type of use case (harsher penalties for dying than taking damage).
* **Respawn**
  + **Description:** If the character dies and still has any lives remaining, the character will be respawned at the beginning of the game.
  + **Step-By-Step Description:**
    - **Extends** Player Death
    - **Includes** Losing a Life
    - The system checks upon death if the character has any remaining lives.
    - If the character has remaining lives, it is placed at the beginning of the game with full health and a decreased score.
* **Losing a Life**
  + **Description:** When the character dies, one of its extra lives is lost, limiting the amount of times the character can die before a Game Over.
  + **Step-By-Step Description:**
    - **Extends** Player Death
    - Upon death, the system checks if the player has any extra lives remaining.
    - If the player does, the system removes one of those lives.
* **Game Over**
  + **Description:** When a player runs out of extra lives by dying too frequently, the game ends and presents the player with the option to restart the game from base settings.
  + **Step-By-Step Description:**
    - **Includes** Restart
    - The system checks, upon death, if the player has any remaining extra lives.
    - If not, the game ends and offers the player the chance to restart the game.
* **Restart**
  + **Description:** Allows the player to immediately restart the game from base settings from the Game Over screen.
  + **Step-By-Step Description:**
    - **Extends** Game Over
    - Detects if the player chooses to restart the game after receiving a Game Over status.
    - If yes, the player character respawns at the beginning of the game.
    - If no, the player returns to the opening/exits the system.
* **Collect Pickups**
  + **Description:** Allows the player to pick up beneficial objects throughout the game.
  + **Step-By-Step Description:**
    - **Includes** Boost Health
    - The system checks if the player is touching a beneficial object within the game.
    - If yes, the object takes effect on the player and the player’s score is increased.
    - If no, the character and object states remain the same.
* **Increase Score**
  + **Description:** Increases the player’s score for actions such as picking up beneficial items and finishing the game in a timely manner.
  + **Step-By-Step Description:**
    - **Extends** Collect Pickups
    - The system checks if any of the criteria (collecting pickups or finishing the game in a timely manner) have been met.
    - If so, increase the player’s score appropriately.
    - If not, do not affect the player’s score.
* **Enter High Score**
  + **Description:** Tells players if they have the highest recorded score on that version of the system.
  + **Step-By-Step Description:**
    - The system checks if the player’s score is higher than the current high score.
    - If yes, inform the player and store that score.
    - If not, do nothing.
* **Complete Level**
  + **Description:** When the character reaches the end of a level within the game with positive health, that level disappears.
  + **Step-By-Step Description:**
    - The system checks if the character is at the endpoint of the level or game and if the character has a positive health.
    - If so, end the level.
    - If not, do not end the level.
* **Enter Best Time**
  + **Description:** Tells players if they have the best/shortest recorded time on that version of the system.
  + **Step-By-Step Description:**
    - The system checks if the player completed the game in the least amount of time compared to previously recorded time(s).
    - If yes, inform the player and store that time.
    - If not, do nothing.
* **Enter New Level**
  + **Description:** Allows the player character to enter a new level once another level has been successfully completed.
  + **Step-By-Step Description:**
    - **Extends** Complete Level
    - The system checks if there is another level to load the player character into.
    - If yes, begin that level.
    - If not, end the game.
* **Win Game**
  + **Description:** Once the player completes all levels with a positive health and at least one life left, the game is over and the player has “won.”
  + **Step-By-Step Description:**
    - **Extends** Complete Level
    - The system checks if there is another level to load the player character into.
    - If yes, begin that level.
    - If not, end the game and inform the player that they have won.