Northeastern

Programming in C++ Project Proposal

Jason Fiammetta, Jerry Leung, Max Mindel, Kimberly Nguyen

February 20, 2018

I. Description

We will be building an original 2D game using OpenGL, OpenAL, and ncurses. The user will be able to control a character using keyboard input and navigate through a generated map world. The user will be able to save, restart, and load the game. The game will use sound cues to tell the user when good or bad things happen. The game will be endless.

II. Target Audience

Our game will be targeted toward all game lovers, particularly those who remember old platformers fondly.

III. Functional Requirements

- The user MUST be able to move the character via keyboard input.
- The game MUST be able to detect collisions between entities and obstacles.
- The game MUST make use of infinite scrolling as the user travels.
- The game MUST automatically generate random terrain.
- The game MUST play music and sound effects in response to events in the game.

IV. Non-Functional Requirements

- The user is able to elegantly leap over obstacles.
- The velocity simulates a real-life environment.
- The music and environment are immersive.
- The menus and controls are intuitive.

V. Use Cases

Use Case:	Gamer creates an account
Primary Actor:	Gamer: a person who likes to play games
Goal in Context:	To create an account for the game so that the gamer can return to a saved character/game state
Preconditions:	● The gamer has the game open in a web browser.
Trigger:	The gamer wants to create an account to be able to play the game.
Scenario:	1. The gamer clicks on the "Create a New Account" link. 2. The gamer is redirected to a page with a form that asks for a unique username and a password. 3. The gamer enters a username and password into the username and password input text fields, respectively. 4. The gamer clicks the "Create Account" button at the bottom of the form. 5. On success, a message appears informing the gamer that her account has been created.
Exceptions:	● The gamer clicks the "Create Account" button without entering a username and/or password. An "invalid username and/or password" error message should be displayed. ● The gamer chooses a username that is already taken. An "invalid username and/or password" error message should be displayed.
Priority:	Low (being able to save the game would be a nice addition, but it is not necessary to be able to play the game)
When Available:	Once the game itself has been completed, if time permits
Channel to Actor:	Web interface (gamer will be using a web browser)
Secondary Actor:	N/A
Channels to Secondary Actors:	N/A
Open Issues:	● How to set-up a database to maintain usernames and passwords ● How to validate usernames and passwords

Use Case:	Gamer creates a character
Primary Actor:	Gamer: a person who likes to play games
Goal in Context:	To create a character with which the gamer will use to play the game
Preconditions:	● The gamer has the game open in a web browser. ● The gamer has logged into a registered account.
Trigger:	The gamer wants to create a character with which she will use to play the game.
Scenario:	1. The gamer clicks the “Create a New Character” button. 2. The screen displays several pre-designed characters from which the gamer can choose. 3. The gamer chooses a character from the available options and clicks the “Next” button at the bottom right. 4. The gamer enters a name for the character into the name input text field that appears. 5. The gamer clicks the “Create Character” button at the bottom right. 6. A message appears informing the gamer that the character has been successfully created.
Exceptions:	● The gamer clicks the “Next” button before choosing a character. An error message should be displayed, telling the gamer to choose a character.
Priority:	High (the gamer needs a character in order to play the game)
When Available:	Within the first two weeks
Channel to Actor:	Web interface (gamer will be using a web browser)
Secondary Actor:	N/A
Channels to Secondary Actors:	N/A
Open Issues:	● How to generate characters using OpenGL ● How to save characters per account

Use Case:	Gamer plays the game
Primary Actor:	Gamer: a person who likes to play games
Goal in Context:	To play the game
Preconditions:	● The gamer has the game open in a web browser. ● The gamer has logged into a registered account. ● The gamer has created a character.
Trigger:	The gamer wants to start playing and hits the “Start” button.
Scenario:	1. The gamer clicks the “Start” button in the middle of the screen. 2. The gamer is directed to the new game. 3. The gamer uses the keyboard controls to run or leap around the generated map. 4. The gamer earns points throughout the game by completing certain objectives (e.g., defeating an obstacle). 5. The game continues until the gamer decides to restart or save and exit the game.
Exceptions:	● The gamer loads a nonexistent game. An error message that says, “The game you are looking for does not exist” should be displayed.
Priority:	Top (this is essentially the entire project)
When Available:	Each aspect of the game will be completed within a certain time frame (see timeline below)
Channel to Actor:	Web interface (gamer will be using a web browser)
Secondary Actor:	N/A
Channels to Secondary Actors:	N/A
Open Issues:	● How to design a game that supports all functional and as many non-functional requirements as possible.

VI. Vision

Our vision for this game is a polished, immersive, and thrilling platformer in which the user must battle relentlessly against obstacles and hazard with increasing speed and challenge. The controls will be intuitive and responsive, and some subset of the additional features listed below will be incorporated to keep the game interesting and allow the user to make meaningful progress from run to run.

VII. Timeline

1. Create demonstrable use of libraries (due 02/20/18)
2. Build a movable character
3. Enable jump physics and platform collision
4. Add hazards with hitboxes
5. Enable window scrolling
6. Create platform-generation algorithm
7. Create sprites for platformer, characters, and hazards
8. Add progress-dependent difficulty/speed
9. Add menu/in-game options
10. Assess progress, and add following features if plausible:
    a. Checkpoints/theme shifts
    b. Save and load functionality
    c. User customization/unlockables
    d. Permanent/temporary upgrades