2D Game Pitch

Grace Murray “Hopper”

CIS 487

November 5, 2016

Joe Starr & Ali Dashti

Version No. 2.0

Project Document Revision History

Version Number	Date	Revision Author	Description of Revision
.01	9/26/2016	Joe Starr	Initial Revision
.1	10/3/2016	Joe Starr	Completed all sections except for 2.1. Ready for review
0.1.1	10/7/2016	Ali Dashti	Review, Editing & Suggestions
0.1.2	10/18/2016	Ali Dashti	Review, Completed Section 2.1
0.1.3	10/18/2016	Joe Starr	Edits on section 2.2,3.3 and review on 2.1
0.1.4	10/19/2016	Ali Dashti	Edits 2.1 & Review
1.0	10/19/2016	Joe Starr	Finished main draft
2.0	11/5/2016	Joe Starr	Final Edits

1 Executive Summary

1.1 Abstract of game story

We will be building an arcade style platformer game. Our game will be a representation of “debugging” code. In our game we will have “code platforms” with bug collectables that must be removed as the player progresses. Our game will center around Grace Murray Hopper. Grace hopper was a Rear Admiral in the United States Navy, she is responsible for the development of compilers, and is credited with coining the term “Bug” for use in computer programming. We will play on all of three of these facts in our game. The game character will be Hopper herself, who will be continuously hopping, the platforms in the game will be the code that is compiling, and the player will score points by removing bugs from the code as it compiles.

2 Game Play Look and Feel

2.1 Appearance

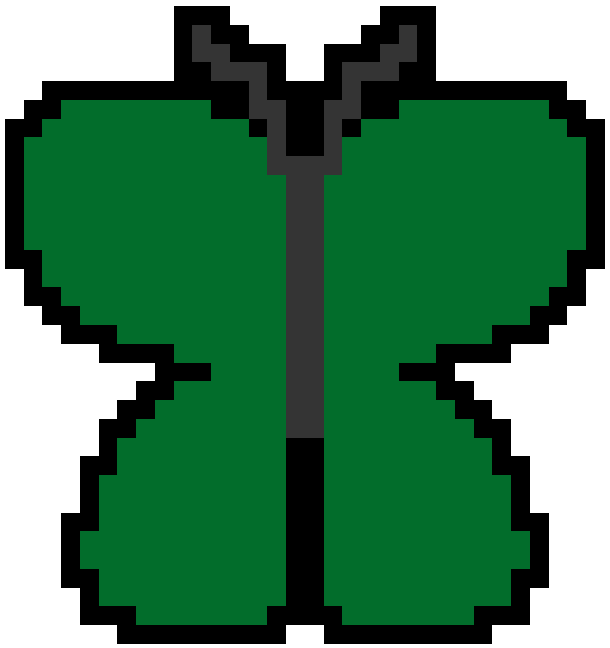
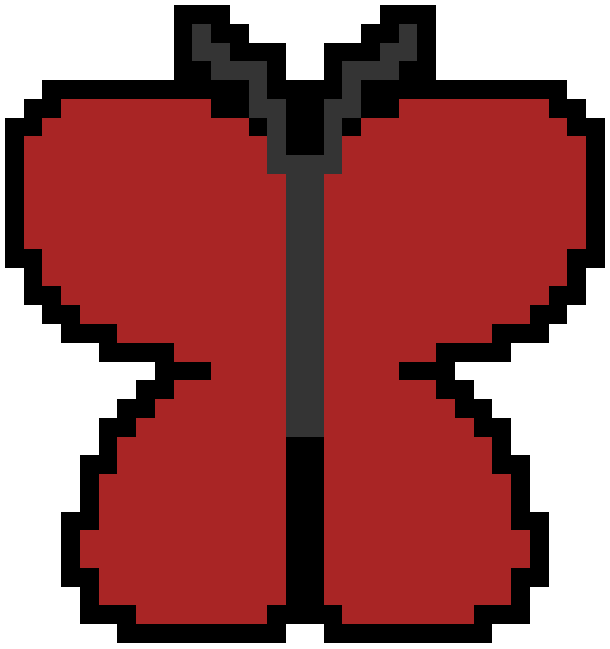
2.1.1 Game Elements

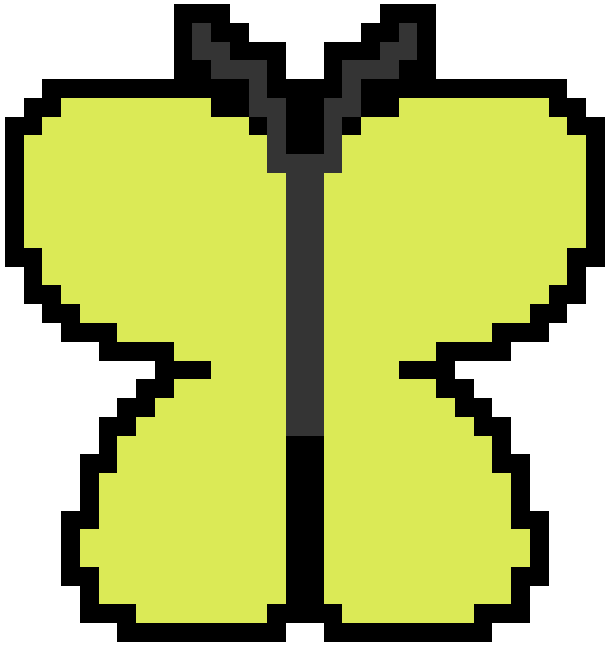
-  Grace Hopper avatar
- Code platforms
  - 

  - 

- 

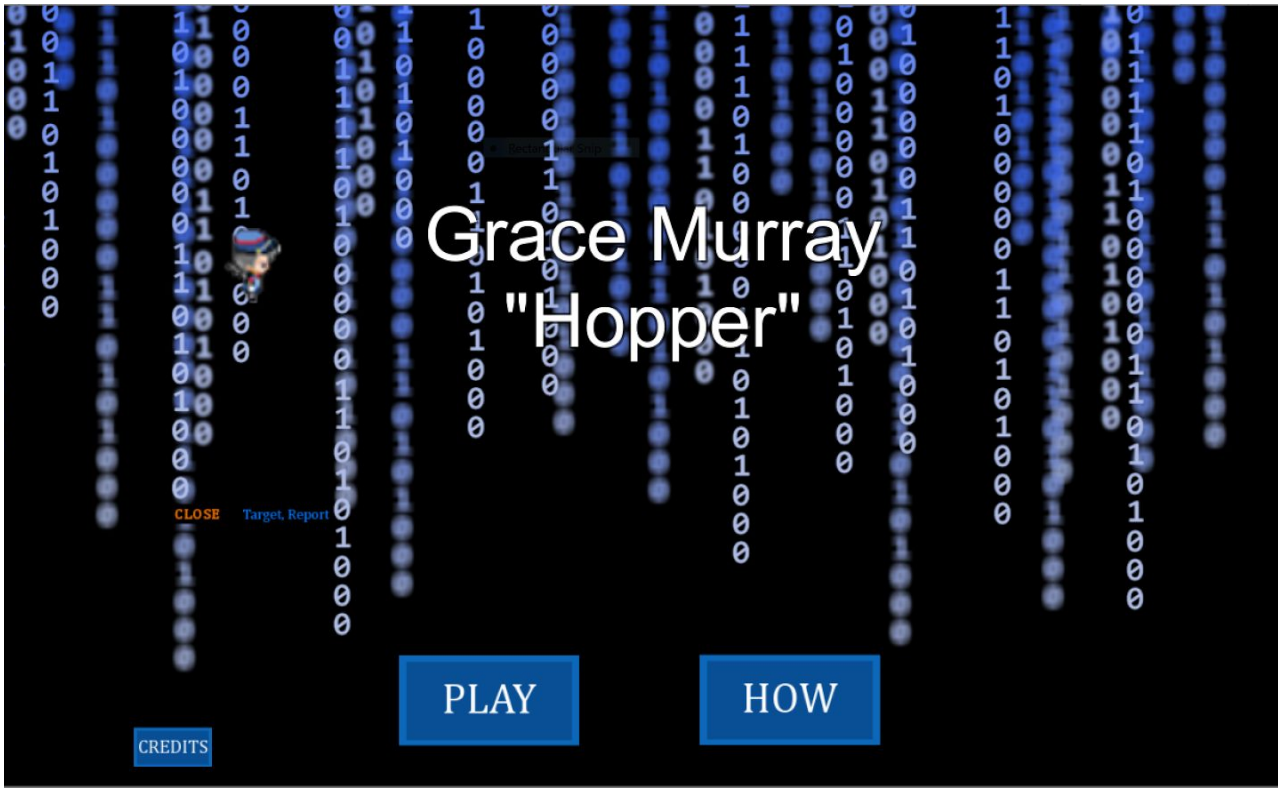
- 

- Bugs
  - 
  - 

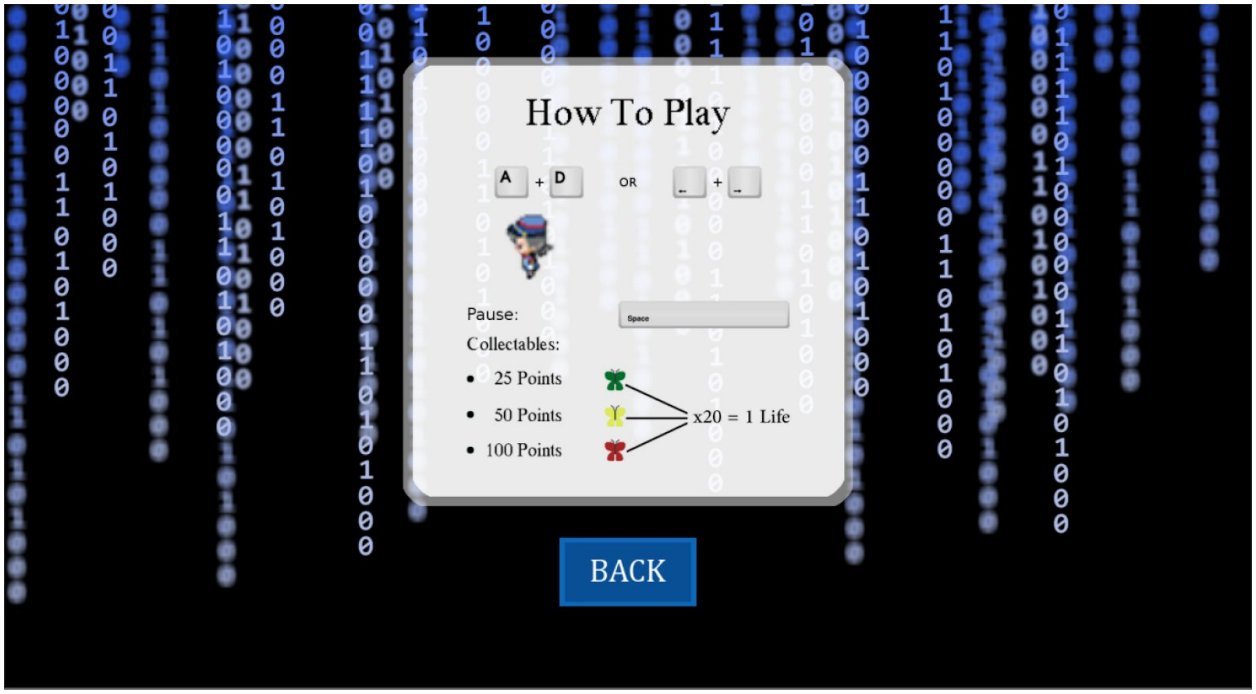


2.1.2 Screen Mock-ups

Main Menu:

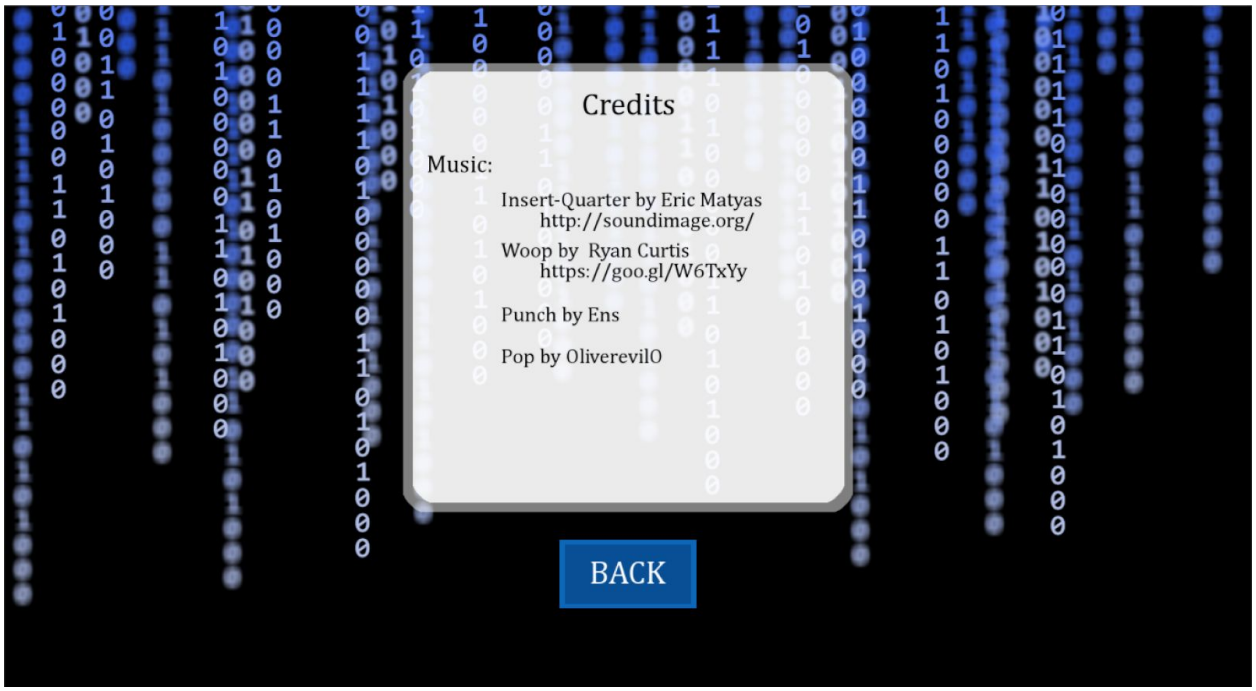


Help Screen:

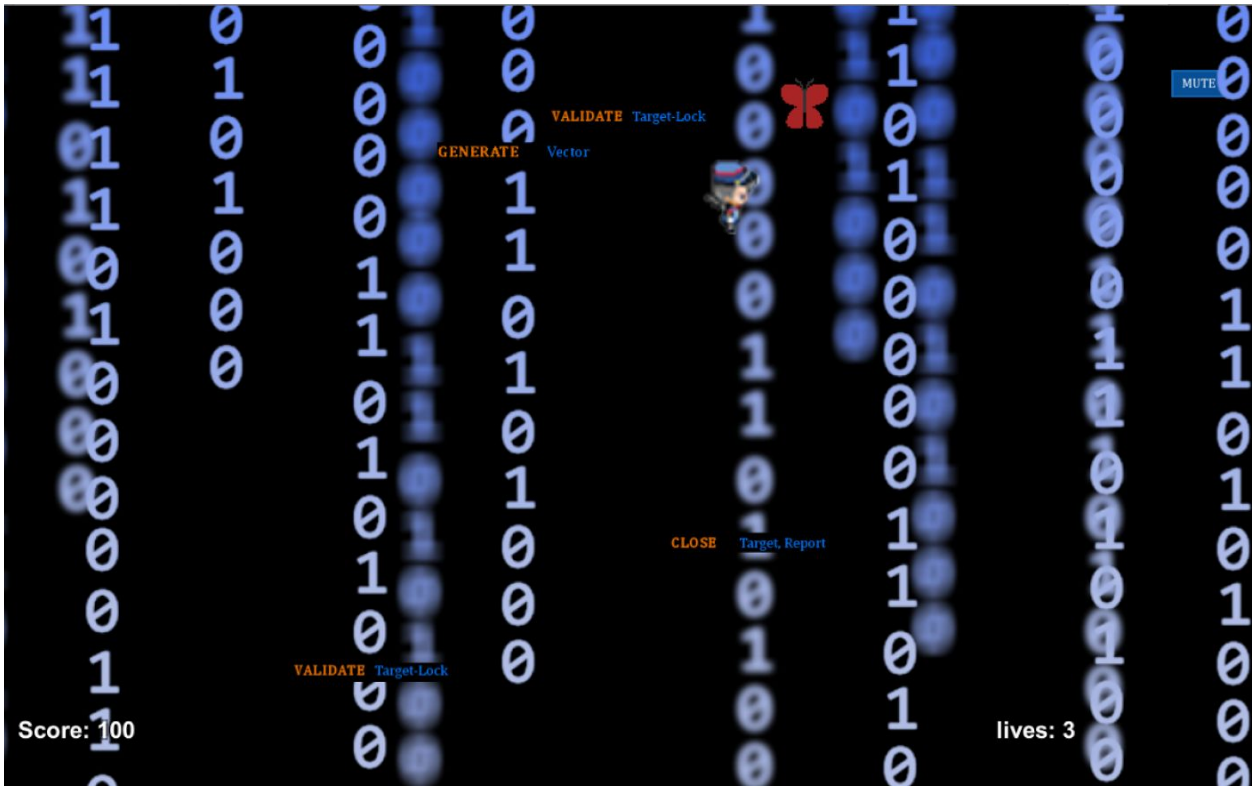


Note: The avatar will automatically move left and right.

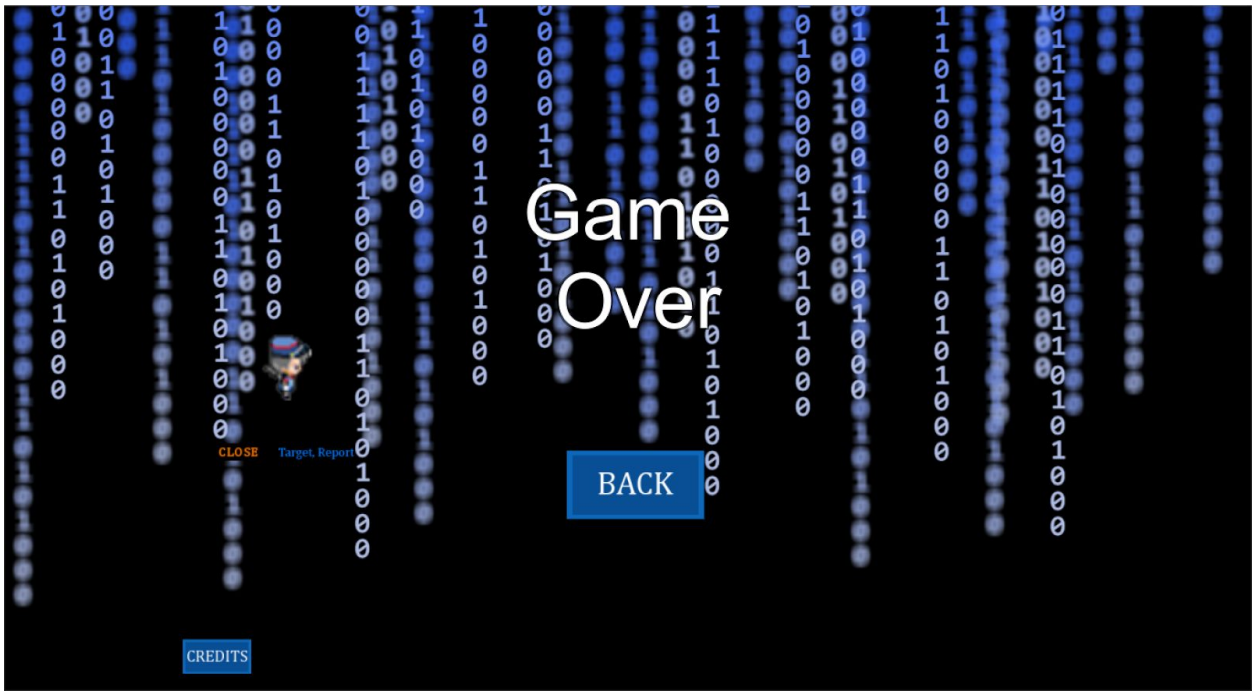
Credits:



In Game Screen



Game Over:



2.2 Player roles and actions

2.2.1 Player Role

The player will control a continuously hopping avatar representation of Grace Hopper. The avatar must ascend “code platforms” that are scrolling down the screen from top to bottom. While ascending the platform the player will score points by collecting bugs they will be scored as:

- Green Bug - 25 points
- Yellow Bug - 50 points
- Red Bug - 100 points

The player is penalized when bugs are missed by losing points based on the type of bug missed:

- Green Bug - 50 point deduction
- Yellow Bug - 100 point deduction
- Red Bug - 200 point deduction

Once the player has collected 20 bugs they will earn a life. If, during gameplay, the avatar makes contact with the bottom of the screen the player will lose a life. When there are no lives remaining the game is over and the player’s final score is the number of bugs collected.

2.2.2 Player Actions

The player will start a game by pressing start on the main menu. The player may also select to view the help menu. During gameplay the player will control the Grace Hopper avatar by using the arrow keys or the A and D keys on a keyboard to move the avatar left and right. The horizontal velocity will change on a button press regardless of if the avatar is in contact with the ground. The player will have no control over "hopping", it will be automatic when the avatar comes in contact with a platform. The player's goal is to collect bugs with the avatar while not falling from the platforms. The player can pause the game by pressing the Spacebar. To resume the player will press space again. The play may also mute the in game audio with the mute button.

2.3 Strategies and motivations

The player must judge based on their intuition the best path to ascend the falling platforms. While they find the best path the player must also maximize a path that contains the most bugs to collect. The player's score is determined on the number of bugs collected, this motivates the player to take risks to gather bugs during gameplay.

2.4 Level summary

There will be only one level in the game. The level will consist of procedurally placed platforms falling from the top of the screen to the bottom of the screen. In addition there will be procedurally generated bugs, also falling from the top of the

screen to the bottom of the screen. The player will be placed on top of the lowest spawned platform.

3 Development Specification

3.1 Hardware

3.1.1 Game Runtime Hardware Requirements

The game will be required to run on a desktop with the following minimum specs (taken from the unity site):

- **Graphics card**: DX9 (shader model 3.0) or DX11 with feature level 9.3 capabilities.
- **CPU**: SSE2 instruction set support.

It is an exciting requirement to have the game run on touch sensitive mobile devices. Mobile support minimum specifications if exciting requirement is implements (taken from the unity site):

- **Android**: ARMv7 (Cortex) CPU with NEON support or Atom CPU;

3.1.2 Game Development Hardware

The game will be developed on a machine meeting the following minimum specifications (taken from the unity site):

- **GPU**: Graphics card with DX9 (shader model 3.0) or DX11 with feature level 9.3 capabilities.

3.2 Software

3.2.1 Game Runtime Software Requirements

The game will be required to run on a desktop with the following minimum specs (taken from the unity site):

Desktop:

- **OS**: Windows XP SP2+, Mac OS X 10.8+, Ubuntu 12.04+, SteamOS+.

Mobile support minimum specifications if exciting requirement is implements (taken from the unity site):

- **Android**: OS 2.3.1 or later;
- **iOS**: iOS 7.0 or higher.

3.2.2 Game Development Software

The game will be developed in Unity 5.4 on a machine running (taken from the unity site):

- **OS**: Windows 7 SP1+, 8, 10; Mac OS X 10.8+.

3.3 Algorithm Style

The game will utilize the unity default gravity and projectile physics to control the motion of the avatar. The motion of the scrolling elements will be governed by vertical scrolling of game elements down the screen. Objects will be held in a list and updated on each refresh of the screen. Bugs will be generated based on a random time, then in Platforms will be generated based on a random time between bugs.