CS 469
Summer 2017

Cassiopeia: Project Plan

Team Members

Bilal Saleem
Robert McGuigan
Monica Pineda

Introduction

Our team aims to create a 2D platformer similar to Mr. Gimmick, a 1992 NES title developed by Sunsoft. This game will be implemented entirely within the game-making engine Unity. Completing this project will give our team a fast-paced introduction and practice in working with Unity and C#, as well as in working through the full software development lifecycle.

User Perspective

Our objective is to make a game that entertains the user, by challenging them to successfully traverse the game's levels while providing them with pleasing graphics and sound. The target user will have already played a 2D platformer, and so will be familiar with running, jumping, fighting, etc. in a 2D side-scrolling environment.

Client

Our instructor, Benjamin Brewster, will be playing the role of our client. The proposed client's requirements are listed below, pending approval.

Client Requirements

- The game will be a side-scrolling 2D platformer.
- The main character will have abilities similar to those of the main character in Mr. Gimmick. Most notably, our character will be able to throw a star at enemies, and this star will bounce around the stage when thrown. The character will be able to ride this star as a platform, reaching areas that would be otherwise inaccessible.
- The game will be created using the game-making engine Unity, with scripts written in C#.
- The game will consist of 4 levels, with a boss at the end of the fourth level.
- The 4 levels will have the following themes: Factory, Seaside, Cave, and Forest. Each level will have some objects and enemies unique to it, as detailed below in the "Game Objects" section. There will also be some objects common to all levels.
- Each level will have graphics and music suited to its theme.
- Each level will have an optional hidden power-up which, if found by the player, will make the remainder of the game easier to complete.
- There will be a “credits screen” displayed after the player completes the game.

Sources:

- We will use pre-made sprites, images, music and sound effects whenever possible, although we may consider creating our own assets if we cannot find something we need.
- Fair use: This software is for educational purposes only and has no commercial aspects to it and it is thereby able to make limited use of proprietary imagery and sound under the “fair use” provisions of the copyright act. All sources will be credited.

Initial Plan

scenes/levels:

Main Menu

- Start Menu
  - Load Game
  - New Game
  - Select Level (Level Map)
  - High Score
    - Loads 5 highest
- Exit
  - Confirm Quit
- Options Menu
  - Instructions
  - Mute
  - Save and Quit

- Factory Level
- Seaside Level
- Cave Level
- Forest Level(boss level)
- End Credits

Game Objects:

Control Keys:

- s=squat
- d=right
- a=left
- spacebar= jump
- j=throw
- p=pause
- Up in the pause menu (mouse)

Mr. Gimmick:

- Actions:
  - Jump
  - Squat
  - Throw star/ magical object
  - Grow big
  - Walk
  - Die
  - Run

Objects:

- Growing mushroom
- Coins
- Levers
- Switches
- Convayer
- Gears
- Clouds
- Rocks
- Giant Tree
- Throwing star/magical object

Enemies:

- Global:
  - Spikes
    - If the player touches the spikes, the player is killed
  - Blobs
    - Always attempt to walk towards the player
    - Deal damage on contact
    - Can be defeated by any of the player's weapons
- Factory-exclusive
  - Crushers
    - Move up and down, touching the ground at their lowest point
    - If the player is caught between a crusher and the ground, the player is killed
    - Cannot be destroyed
  - Robots
    - Move back and forth on a single platform, looking in the direction they are moving

            - If a robot sees the player, it fires some bullets in a straight horizontal line, which do damage to the player on contact
            - Can damage the player by direct contact
            - Can be defeated by any of the player's weapons

    - Turrets
        - Periodically fire a single bullet directly at the player, which does damage on contact
        - Can be defeated by any of the player's weapons

- Seaside-exclusive
    - Seagulls
        - Fly quickly in a straight horizontal line, against the player's path, causing damage on contact
        - Can be defeated by any of the player's weapons

    - Flying fish
        - Jump (vertically? diagonally?) out of bodies of water at the bottom of the screen, causing damage on contact
        - Can be defeated by any of the player's weapons

    - Crabs
        - Moves fast back and forth on a single platform
        - Can damage the player by direct contact

- Cave-exclusive
    - Turtles
        - Move back and forth in a single platform
        - Once the player hits them their shells become a weapon moving faster back and forth in the platform
        - Can damage the player by direct contact
        - Can be defeated by direct contact or player's weapons

    - Lemmings
        - Jump up out of spaces that the player needs to jump over
        - Can damage the player by direct contact

- Forest-exclusive
    - Giant Robot Squirrel
        - Throw/drops nuts on the player, causing damage
        - Can be damage the player by the player being hit by the nuts or direct contact
        - Can be defeated by the player's weapon or direct contact

Game Manager:

- Each level inherits from the parent class
- Contains:
    - Health points
    - Current score
    - Displayed score (score shown on screen, counts upwards towards “Current score” whenever points acquired)
    - Audio Manager
    - Score Manager
    - Items Accumulated (count of coins)
    - Persistence Script
        - Player Object
        - HighScore Object
        - Save Player File
        - Load Player File
        - Save HighScore File
        - Load High Score File

File for GameSave:

- Saves preferences above
- Saves top ten high scores
- Save status
- Load status
- Present option to save at end of each level

File for High Score Table:

- Top 5 based on calculations
- Score updated during level
- Score Calculation(TBD)

Folder Structure:

- Animation - animations and animation controllers
- Audio - sound files
- Font - text fonts
- Materials - templates to describe shading / color etc that can be added to an object.
- Physics Materials - physics description of an object e.g. friction, bounciness
- Prefabs - templates / groupings for layout items and game objects
- Scenes - different level / menu layouts
- Scripts - C# scripts for game

- Sprites - 2D images (appropriated otherwise custom made)
- Textures - non sprite images

Game Objects Unique to Each Level:

- Unique Backgrounds
- Unique Wall / Floors
- Special Items (such as falling platforms, falling boulders, etc)
- 2-3 Unique enemies or threats
- Unique Environmental threats (such as lava, falling to death, etc)

Level Testing:

- General Physics testing
- Individual Level Testing
- Integration Testing

Software:

- The Unity game development engine
- C# for scripting

Minimum Requirements to Run Finished Game 1 :

- OS: Windows XP SP2+
- Graphics card: DX9 (shader model 3.0) or DX11 with feature level 9.3 capabilities.
- CPU: SSE2 instruction set support.

Team Member tasks

Robert McGuigan :

Task	Time Estimate (hrs)
Week 3: Work on predetermined set of Global Items Work on predetermined set of Global enemies Loading and saving data Continue to plan Factory Level	16
Week 4: Complete predetermined set of Global Items Implement team members global Items and Enemies Complete persistence data Complete Factory Level planning Start on backgrounds/floor/wall for Factory Level Video progress report	14

Week 5: Start and complete player /enemy programing unique to Factory Level Start and complete collision interactions for Factory Level Produce mid-point check materials	30
Week 6: Complete rough Factory Level All Level Testing Start Forest Level(as a team)	15
Week 7: Finish Forest Level (as a team) Categorize and Fix high priority bugs and updates from testing	22
Week 8: Final report Input Demo File Submission	8
Total Time	**105**

Monica Pineda:

Task	**Time Estimate (hrs)**
Week 3: Work on predetermined set of Global Items Work on predetermined set of Global enemies Integrate Persistence Work on saving and loading data Planning of Seaside Level	15
Week 4: Complete predetermined set global Items and Enemies Implement team members global Items and Enemies Complete persistence data Complete Seaside Level planning Start on backgrounds/floor/wall for Seaside Level Start on Player/Enemy interactions for Seaside Level	17
Week 5: Start and complete player /enemy programing unique to Seaside Level Start and complete collision interactions for Seaside Level Produce mid-point check materials	30
Week 6: Complete rough Seaside Level	15

All Level Testing Start Forest Level(as a team) Video progress report	
Week 7: Finish Forest Level (as a team) Categorize and Fix high priority bugs and updates from testing	22
Week 8: Final report Input Demo File Submission	8
Total Time	**105**

Bilal Saleem:

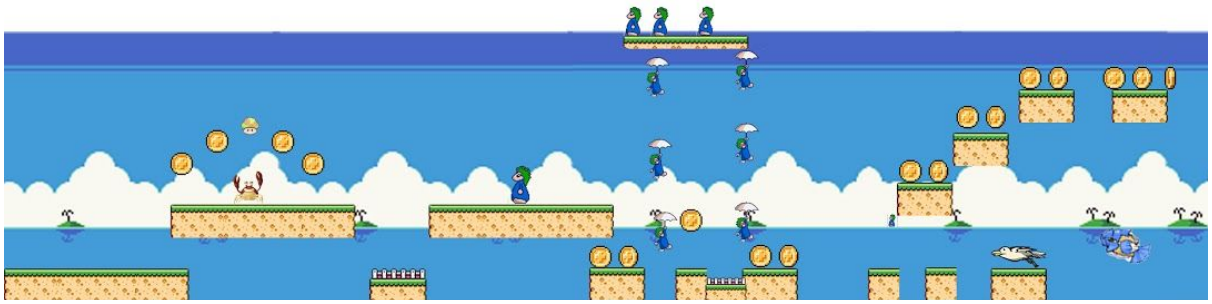
Task	**Time Estimate (hrs)**
Week 3: Work on predetermined set of Global Items Work on predetermined set of Global enemies Integrate Persistence Work on saving and loading data High score table generation	15
Week 4: Complete predetermined set global Items and Enemies Implement team members global Items and Enemies Complete persistence data Complete Cave Level planning Start on backgrounds/floor/wall for Cave Level Start on Player/Enemy interactions for Cave Level	17
Week 5: Start and complete player /enemy programing unique to CaveLevel Start and complete collision interactions for Cave Level Produce mid-point check materials	30
Week 6: Complete rough Cave Level	15

All Level Testing Start Forest Level(as a team)	
Week 7: Finish Forest Level (as a team) Categorize and Fix high priority bugs and updates from testing Video progress report	22
Week 8: Final report Input Demo File Submission	8
Total Time	**105**

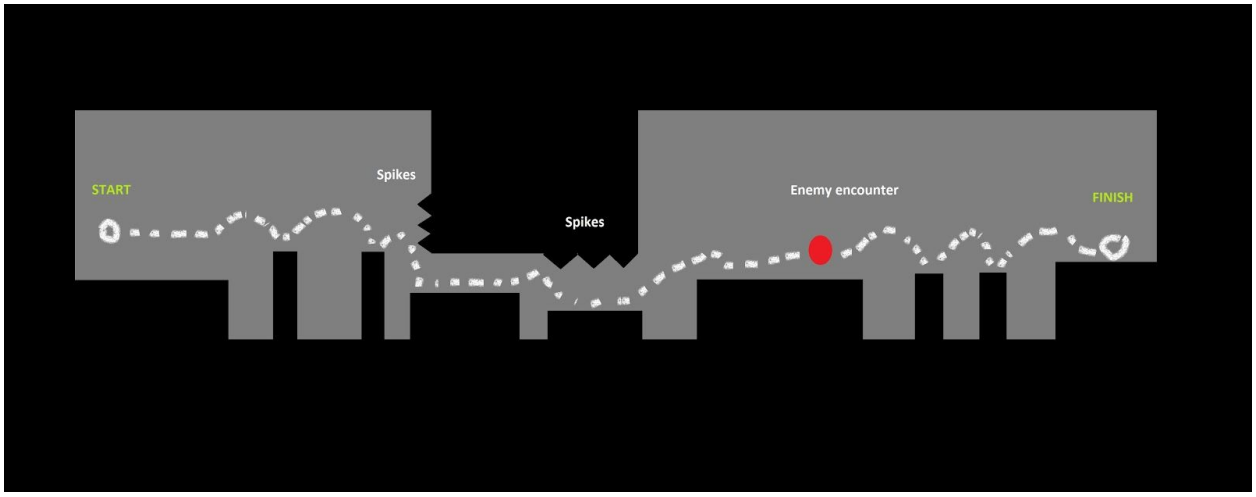
Repot	**Due**	**Format**	**Assignee to Submit**
Project Plan	July 6	Written Plan	All
Week 4 Progress Report	July 17	Video	Robert
Mid-point Project check	July 24	All files + written report	All
Week 6 Progress Report	July 31	Video	Monica
Week 7 Progress Report	August 7	Video	Bilal
Final Report	August 18	Written	All
Demonstrate Project	August 18	All files Zipped	All

Graphical Examples:

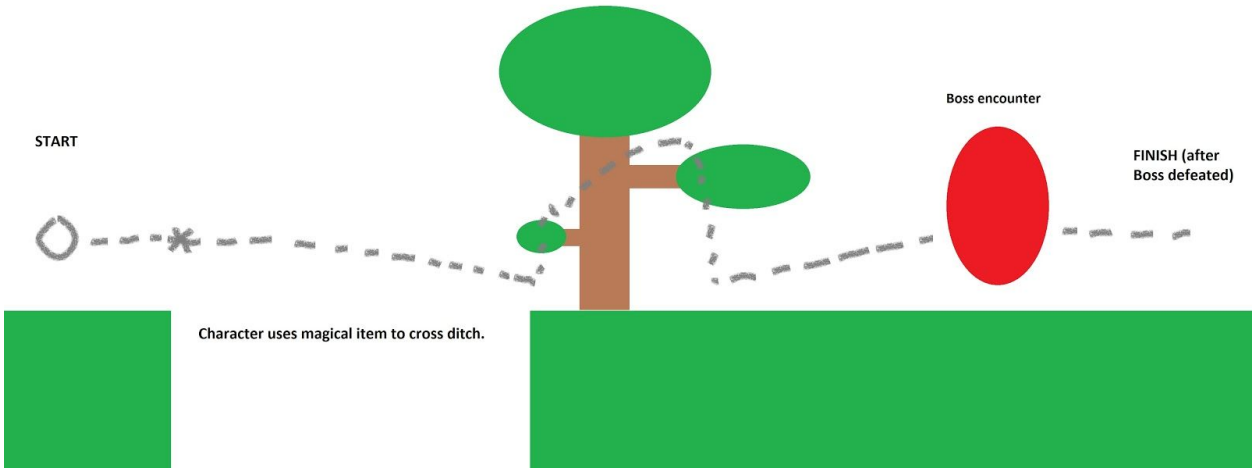
Seaside level



Cave Level



Forest Level



Conclusion:

The Cassiopeia team hopes to create a side scrolling 2d platformer game by making use of features of the Unity game engine, managed through scripting in c#. The project should take at least 300 hours to complete and should be completed on schedule.

References:

[1]https://www.udemy.com/unity2dplatformer/
[2]https://unity3d.com/unity/system-requirements
[3]https://unity3d.com/learn/tutorials/topics/scripting/persistence-saving-and-loading-data
[4] Hula Project Plan (Sample Project Plan Provided by the professor)