Final Milestone: The Little Prince & the Lost Rose

CPSC 427 - Video Game Programming

Fall 2019/20

Team Members

Name	Student ID
Grace Chang	36569151
Margaret Yao	50945161
Cheryl Yao	35939157
Stephanie Liu	25319147
Yue Deng	56890163
Yang Muxi	87191680

Story

Background Story

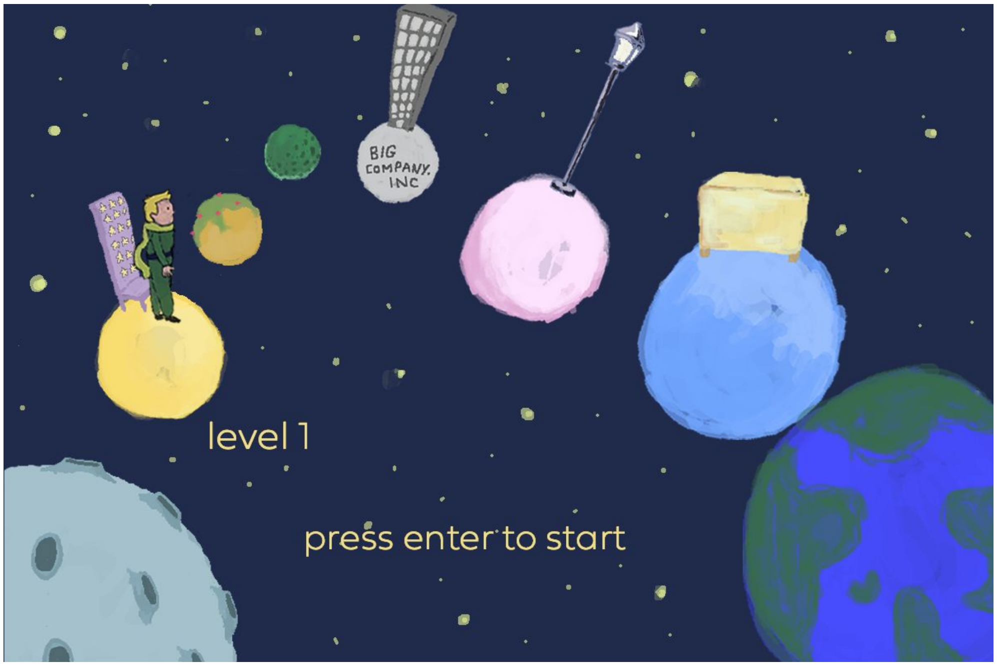
An unknowing astronaut has found the Little Prince’s Rose and took it back to Earth as proof of life in space. Now, the Little Prince has to travel across the galaxy back to Earth in order to get his Rose back!

Motivation

Playing as the Little Prince, you must navigate each planet he visits and find a key to leave the planet and travel onto the next one. As you advance through each planet, the number of enemies increase. You can fight by collecting stars. These stars help you attack the enemies. The game will be RPG, and gameplay will be similar to “Legend of Zelda” NES (1986), in which you fight/avoid enemies and find objects to meet your goal.

Levels Descriptions

Due to time and resource constraints, each level represented each planet via a “theme”. In our final game, we made sprites according to our game proposal. However, we decided to take out the “Earth-First Landing” level as the Little Prince was travelling planet to planet, and in between each level we’d made a visual indicator for the user to know they are going to play the next level:



If he were to travel within one planet suddenly, the sudden change in this established game flow may confuse the users.

We also changed the objective slightly. After one of our cross-play sessions, we received feedback that the changes between levels were too sudden (prince would get key and suddenly go into the next level). As such, we implemented a portal to indicate to users that something would happen if they went through it. We also implemented the between-level-screen (mentioned & pictured above).

The stars, which acted as a point system, became how many times the prince could attack. This was because as we developed the game, we did not see how a point system would motivate users to collect stars. As such, we decided that the prince would attack using the stars he’d collected.

Planet 1 - The King with Sad Subjects

Enemy Sprites: Palace Guards
Setting: Palace
Attack System: Stars
Objective: Find a golden key and go through the portal to the next planet.

Planet 2 - The Narcissistic Man

Enemy Sprites: A dude looking into a mirror
Setting: Simple Rough Terrain
Attack System: Stars
Objective: Find a golden key and go through the portal to the next planet.

Planet 3 - The Drunk Man Alternative

Enemy Sprites: Drunk Dudes
Attack System: Stars
Setting: Simple Rough Terrain
Objective: Find a golden key and go through the portal to the next planet.

Planet 4 - The Business Man Who Loves Stars

Enemy Sprites: Angry Dudes in suits
Attack System: Stars
Setting: Office Building
Objective: Find a golden key and go through the portal to the next planet.

Planet 5 - The Lamplighter

Enemy Sprites: A man with an oil lamp
Attack System: Stars
Setting: Dark Terrain
Objective: Find a golden key and go through the portal to the next planet.

Planet 6 - The Elderly Geographer

Enemy Sprites: Geographers
Attack System: Stars
Setting: Geographer’s workspace
Objective: Find a golden key and go through the portal to the next planet.

Earth - Space Station

Enemy Sprites: Astronauts
Attack System: Stars
Setting: Space Station
Objective: Get your rose!

Technical Elements

* is where discrepancies from the original game proposal occured
Text in *blue italics* are features we didn't think to add during the proposal, but are now added

Rendering

- 3D illustrations for intro sequences*
- 2D pixelated sprites on a 2D map during gameplay

*Instead of 3D illustrations, we stuck to a 2D narration short video. This was easier to make + manipulate as someone on our team was already well versed in photoshop.

Assets

- The Little Prince will be our main Sprite.
- Enemy sprites for each level.
- Different objects that the little prince will interact with throughout the levels, such as stars.
- *Intro screen, screens to indicate levels, game over screen, won game screen, dead screen*
- *Narration video*
- *Barrels spritesheet*
- *Stars pop-up window*
- *The "ground" of each level*
- *Enemy and prince spritesheets*

2D geometry manipulation

- Throughout the game, different characters and elements will move (transform).* This game design element is also satisfied by how users have to use keyboard buttons to move the Little Prince around.
- Collision is satisfied for when the Little Prince collide with an enemy sprite.
- *Additional collisions included:*
    - *prince collide with portal*
    - *"bullets" (prince/enemy attack) colliding with enemy/prince/barrels/other bullets*
    - *prince colliding with stars, hearts, and wall*

*Although our enemies move in the game, this is done by spritesheets instead of 2D geometry manipulation. However, the Little Prince still moves via 2D geometry manipulation.

Gameplay/logic AI

- Little Prince is controlled by arrow keys
- Logic AI can be satisfied by the random spawning of each enemy the Little Prince has to fight or avoid.

Physics

- Little Prince knockback when collided with sprite*

*Instead of doing this, we implemented physics by adding velocity/acceleration to the prince movement.

Advanced Technical Elements

* is where discrepancies from the original game proposal occured
Text in *blue italics* are additional features that were added

Assets

- Audio for feedback
    - If this is skipped, there may be a chance of having bad feedback. This could be remedied by having visual feedback

Gameplay/logic AI

- Spacebar to attack.
    - If this is skipped, gameplay will be less satisfying, as the Little Prince can only avoid colliding with enemy sprites.
    - To further enrich this gameplay, there can also be different modes of attack. Not too important if skipped.
- Spacebar to advance through the dialogue, and to interact with NPCs/objects*
    - If this is skipped, users will have a less enriched story experience
- Depending on level creation, a more advanced technical element that can be included is procedural generation*[1]
- Health Bar
    - Not too important if skipped -- gameplay may be shortened as player will die more quickly
- *Barrels contain stars within them*
    - *Prince can throw stars at barrels*
    - *Either 0, 1, or 2 stars are contained within each barrel*

Physics

- Enemy knockback when struck by a weapon*[2]

Stretch Goals:
Additional mini level for Planet 3*[3]

* We decided to remove dialogues and interaction with NPC's as a whole
*[1] Instead of procedural generation, level maps were determined by reading text files
*[2] Instead of enemy knockback, we decided to just kill the enemy on hit
*[3] Mini levels were excluded due to time and resource constraints

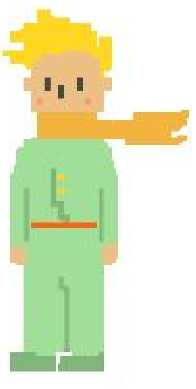
Devices

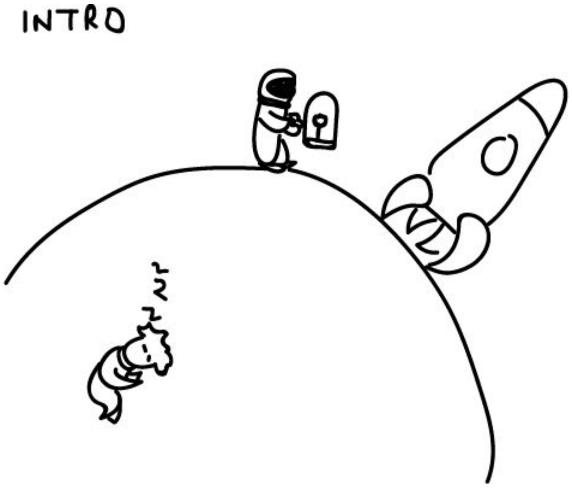
In our original game proposal, we planned on having the game be playable on Windows 7/8/10. However, during development, a lot of Windows specific errors occurred (ie. some vector pointer issues would occur on Windows and not on Mac). Additionally, opencv, an external library we implemented to play our intro narration video, was easier to install on Macs. As such, we changed our main device to be playable on Mac OS X.

Mac OS X

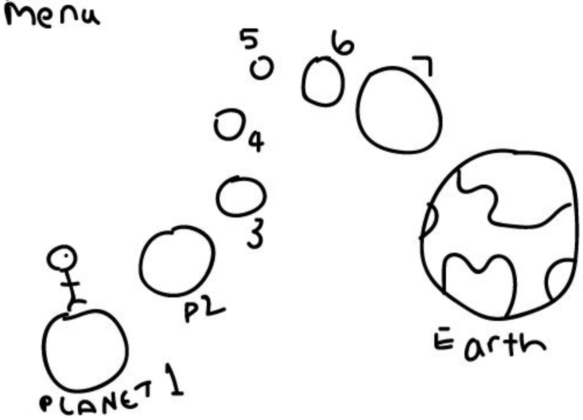
- Arrow keys for control
- Space bar for attack
- H to go to help screen
- S to save game
- N to play new game
- C to continue game from save file
- Esc to quit game or exit help screen

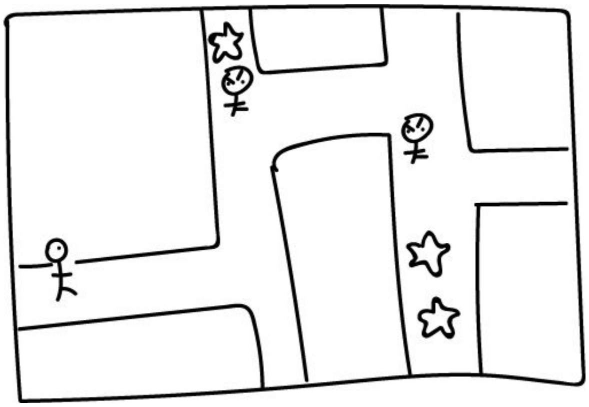
Concepts



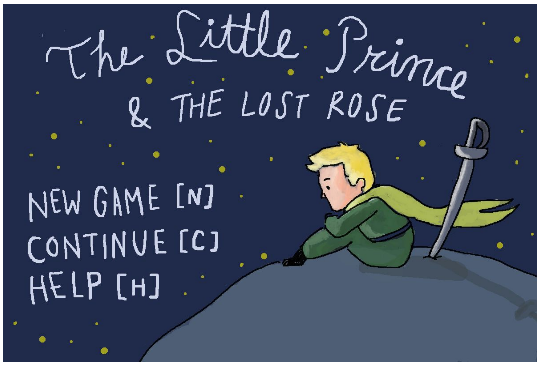


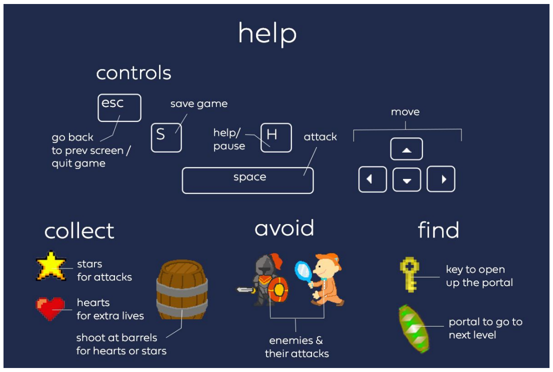
Main Menu





FINAL GAME/ ART





help

controls

esc — go back to prev screen / quit game

S — save game

H — help/ pause

space — attack

move

collect

stars for attacks

hearts for extra lives

shoot at barrels for hearts or stars

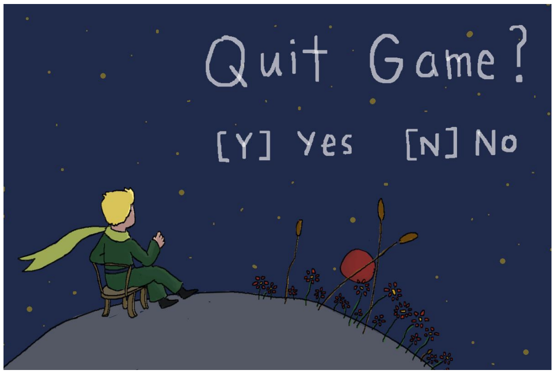
avoid

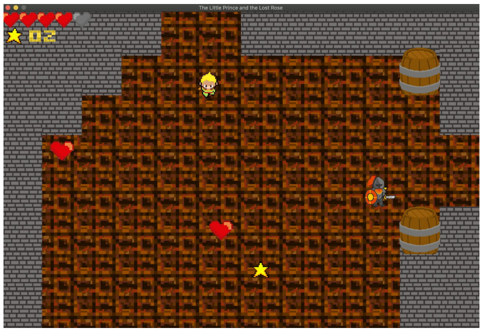
enemies & their attacks

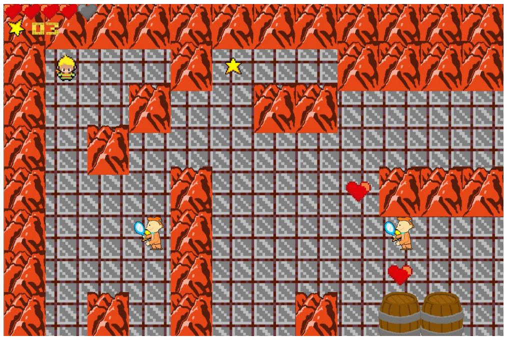
find

key to open up the portal

portal to go to next level







Tools

Although we tried to use Box2D for physics related features during the last milestone, we ultimately went with none of the tools that was in our game proposal document. This was partially due to the fact that we did not have any specific functionalities that would need an external library.

However, when we made the narration video in the last milestone, we needed something that would play the video. As such, the external library we implemented was **opencv**. We used opencv to open, load and play the video frame by frame.

Development Plan

We were mostly able to follow our development plan.

* is where we were unable to implement due to time constraints

Things we did not account for in the game proposal are noted in *blue italics*.

Week: October 4 - Skeletal Game

For the skeletal game, we will be focusing on developing the overarching, generic functionalities of our game. We will be focusing on...

Gameplay such as:

- Player control
- Enemy movement
- Collisions

Developing sprites for:

- Enemy
- Player

We had to find a way to lay out the setting of the game in a way that would not cost memory. We ended up using a txt file to read and map out each tile in our game.

October 11 - End Week 1, Milestone 2

Once we have our skeletal game, we will focus on

- Developing further levels by changing the texture of our enemy sprites
    - Locking/unlocking levels
    - Checking if player has found a key so that they can advance to the next level
    - Save file to save player progress*
- Building out environment of each level

Additionally, time will be allotted to debug or fix anything from the previous week.

For this week, we focused more on building out additional levels. This took some more time as sprites needed to be developed and new textfiles had to be made. We also added in Prince collision with walls. As such, a save file was not made.

October 18 - Minimal Playability (Milestone 2) Due

We will focus on feedback from users, and change our game as needed. If there is not much to change, there is a possibility that we can focus on extending functionalities like:

- Little Prince can shoot at enemies*
- Audio feedback*

Feedback included some collision problems and key problems. Users wanted the Prince to move diagonally if two arrow keys (up & right, down & left, etc) were pressed. Users also noted that it seemed as if the Prince went to the next level too suddenly. We focused on implementing user feedback, as well as debugging functionalities from week 1. As such, we did not extend an attack functionality, nor audio feedback.

Functionalities we implemented for milestone 2:

- *implemented tutorial level*
- *implement condition for not respawning stars when collected after leaving the map and coming back to it*
- *implemented screen state (intro, in-btwn-level)*
- *fixed invisible portal bug*
- *heart bar for health (on diff branch)*
- *star bar for points (on diff branch)*
- *star animation when collected*
- *Refactored code*

October 25 - End Week 1, Milestone 3

We could still be developing changes from user testing. We will begin testing and debugging any changes in game that was made. If there is not much to change, we can begin debugging extended functionality from last week. There is also a possibility that we can focus on extending functionalities like:

- Different modes of attack the Little Prince can have, based on what objects that are picked up*
- Being able to interact with enemy (ie, talk to King, etc)*

We had many bugs that were left over from the previous milestones. As such, we did not extend functionality. Below were some we fixed throughout the week:

- *Hearts, stars and enemies disappear when prince goes off screen and then comes back*
- *Another key is spawned after prince takes it and portal appears, leave the screen and then comes back (two key issue)*
- *If prince goes off screen and comes back repeatedly, key doesn't show up*

- *If prince goes off screen and comes back to screen with key in it, the key respawns in a different position*
- *Key generates on first screen of level, making gameplay too easy*
- *Key will show up next to wall, which makes portal spawn on a wall and make it impossible to advance to next level):*
- *Sometimes enemies spawns on top of walls*
- *Star bar not showing on upper left corner of screen*

November 1 - End Week 2, Milestone 3

Wrap up on testing and debugging any additional changes in our game. We will focus heavily on aesthetic of game and user experience.

While we did focus on user experience (game flow), we did not finish on debugging. We continued to debug, while implementing more levels, generate new audio, and making the main screen and game over screens for the game.

- *Implemented more levels (generate .txt files)*
- *Implemented background sound*
- *Made legit intro screen*
- *Made end screen for when prince dies*

November 8 - Playability (Milestone 3) Due

We will focus on feedback from users, and change our game as needed. If not much is changed, we can extend functionality.

We had to focus on feedback from users instead of extending functionality. Some feedback we received were:

- *put in arrow keys for where prince should advance on the screen*
- *on_key press implementation*
- *fix restart button*
- *prince/wall collision bounceback*
- *track how many of the total amount of stars have been collected (i.e. 1/3 stars collected etc.)*
- *Prince is animated through a spritesheet*

Week: November 15

Debugging any changes made.

Along with debugging, a feature many users in the crossplay wanted to see was exploding barrels. As such, that feature was implemented along with:

- *A better indicator for how many hearts the prince has (capped at 5, grey hearts if the prince has less than 5 lives)*
- *A pop-up window*
- *Animated enemies using spritesheets*
- *A new ground customized for each level*
- *Barrel spritesheets and a pop-up window showing how many stars are collected*
- *Exploding barrel particle system*
- *Updated the tutorial to reflect the new features*

Week: November 22

Additional touches and final debug. We will not be doing any major changes this week.*

By the final milestone we fixed all known bugs and memory leaks. A major change we made this week was integrating opencv into our game to satisfy the external integration requirement. In addition to creating a narration to play at the beginning of the game, we also added:

- *A final win screen and replacing the key with the rose in the final level*
- *Enemies having its own customized bullet*
- *Audio for most game interactions*

Week: November 29 (Extended to December 6) – Final Game Due