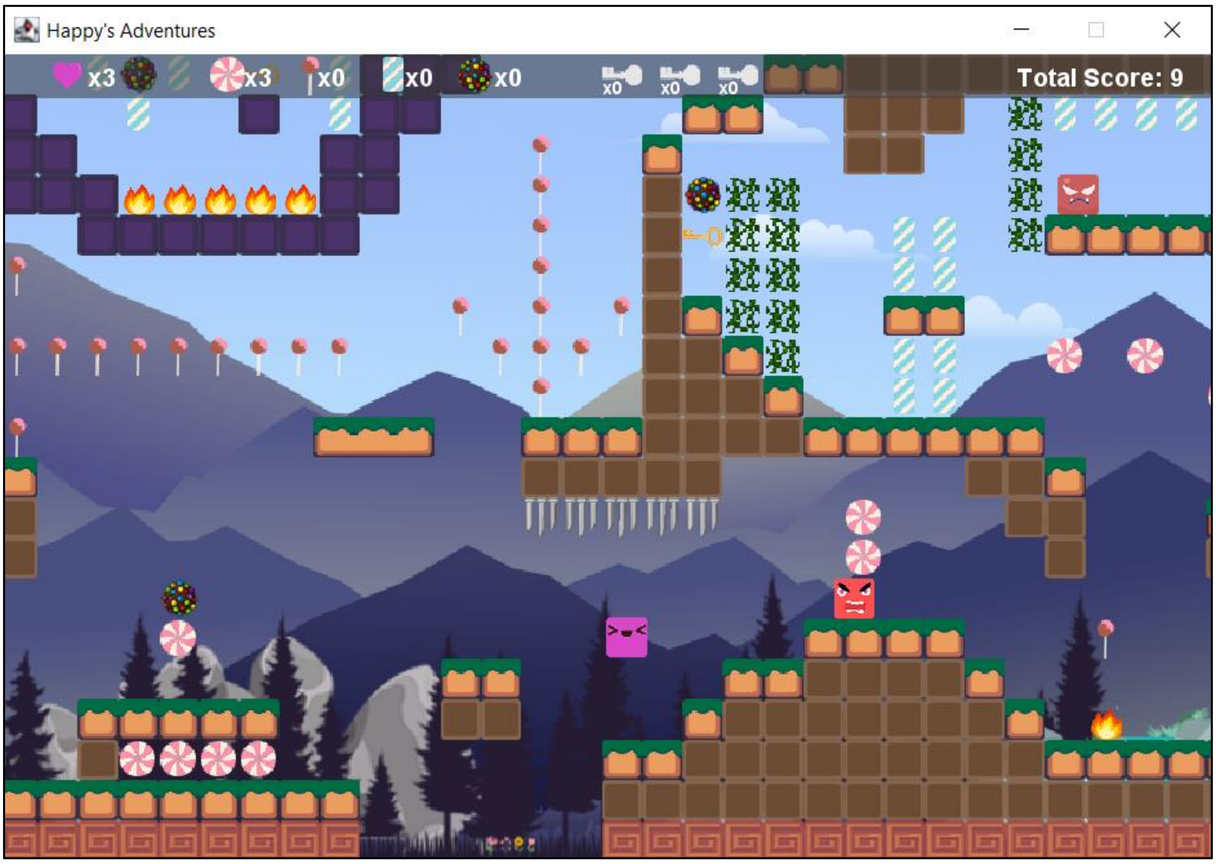
HAPPY'S ADVENTURES

GAME DESIGN DOCUMENTATION



GAME OVERVIEW

Happy's Adventures is a vibrant and nostalgic platform game developed by The Good Gamers. Inspired by the classic platformers of the 80s and 90s, the game takes players on an exciting journey with Happy, the pink square hero, as he jumps on blocks and collects items to save his friends. Developed in Java, the game features colourful blocks, old-school platform mechanics, and a user-friendly approach to level creation using CSV files.

THE STORYLINE

In the vibrant realm of SweetLand, a once-friendly community has fallen victim to a mysterious evil that placed a curse on them and transformed them into wicked minions. Happy's four cherished friends managed to escape the curse but instead have been imprisoned. Now, Happy embarks on a life-defining mission to rescue his friends and break the curse, restoring peace and sweetness to SweetLand. Along the way, he must navigate perilous obstacles and avoid hazards before his friends also succumb to the curse. Can Happy save them all and bring harmony back to SweetLand?

GG The Good Gamers

GAME PLANNING AND DESIGN

Considering the tight deadline and lots of ideas for the game, the team had set up a Miro board to contain every aspect of the game planning phases, ideas and project timelines to keep track of various deliverables throughout the game's development. Everyone had tasks assigned and weekly meetings via Zoom were held to discuss progress and check off the items that have been tasked and completed for each week. Discord was the main channel of communication and the team worked well to communicate any roadblocks and hurdles along the development journey. Link for the Miro board at the end of the document.

GAME FEATURES

COLLISION DETECTION

Using Java programming, the team implemented collision detection logic to ensure accurate interactions between Happy and the game environment. This allowed Happy to stand on blocks, interact with items (hazards, enemies, and interactive objects), and navigate through the world using intuitive keyboard keys. Collisions were implemented creating Java rectangle class so hitboxes could be used to manage the various types of interactions within the game world.

JUMPING MECHANICS

The game's jumping mechanics were crafted to provide responsive and satisfying controls for players. This involved coding simple physics and methods to control Happy's jumping ability, ensuring players a satisfying experience, particularly for those already familiar with platform game mechanics.

MAIN CHARACTER

The main character of Happy's Adventures is Happy, the pink square. The player will control Happy to interact with the various items in the game world, as well as enemies and the obstacles set out to hinder in Happy's quest to rebuild SweetLand to its former glory.

ENEMIES AND HAZARDS

To increase the game difficulty, hazards were implemented into the game (fire, spikes, enemies, and holes) and can cause damage to the character. Enemies have individual movement and can also become friendly if Happy interacts with an evil minion while at least one SuperSweet is in its inventory.

FRIENDS

Friends can be rescued by finding the right coloured keys for the matching locked doors, or by "converting" enemies back into friendly characters. Once Happy interacts with a friendly character, the friendly will continue to follow Happy until they reach the safe zone area (their Home) and they are considered saved.

ITEMS

Candies were used mostly as a scoring system. There are three types of candies in the game world, each attributing a different value towards the total score. The SuperSweet is a special item that allows the curse to be broken on enemies and converts them into friendlies that can be saved. Three different coloured keys are found throughout the game world which is used to unlock doors. Happy will have to find the keys and interact with the same-coloured doors to be able to save his friends and reach the end of the level that is sealed by three locked doors. Happy can also collect hearts which will add one heart to his total hearts. Happy starts with three hearts at the start of the level, Happy loses hearts when he interacts with hazards, and once the total number of hearts reaches 0, the game is over.

USER INTERFACE AND GAME MENUS

The game's graphical user interface was built using Java Swing, which provided the foundation for rendering and handling user input. There are various menus to facilitate user input throughout the different stages in the game, such as Main Menu, Pause Menu, Retry Menu and Quit Menu. When a new game is started, a “How To Play” screen will provide basic instructions for the player to control Happy in the game world. There are a few tutorial popups at the start of the game to help the player understand the mechanics of certain collectables found in the game.

TIMER CONTROL

A timer was implemented to regulate events within the game, such as determining hit time length. Java utilities Timer and timerTask classes were used to achieve that. When the character is hit, he has a two seconds window to flee and avoid hazards. This was done so the players would not constantly lose their health and have time to promptly react to the hazard that hit them. As a result, players can use timing as part of the gameplay experience.

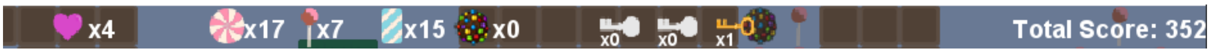
ANIMATION AND SPRITES

A series of methods were created to load sprites to provide the animation to characters and objects. All sprites were custom made so each character from the game could have a unique look and animation. Sprites were added for the game visual assets, enemies, Happy and friendlies.

UTILIZING "GAME ENGINE" FOR DEVELOPMENT

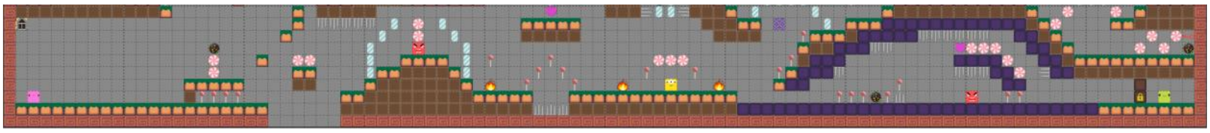
To facilitate development and streamline the coding process, the Game Engine provided a framework with pre-built functions and tools that simplified the implementation of game mechanics, rendering, and other essential components. The engine's features included ready-to-use methods for rendering, audio, framerate handling among others. By using the capabilities of this engine (which was also part of this assignment requirement), the team could focus more on designing the unique elements of Happy’s Adventures while reducing the overall development time.

SCORE SYSTEM



A HUD (Heads up display) at the top of the screen will show a player’s progress throughout the game. This is integrated with the scoring system. The score increases as the player catches different candies scattered throughout the world. The amount of candies collected is also tracked and displayed. The HUD also shows how many keys Happy has collected, and the amount of Supersweets available for Happy to use.

CSV-BASED LEVEL CREATION



The team investigated a series of options to facilitate game level design. One notable feature of Happy's Adventures is the use of CSV files to create game levels. By creating a level using the "Tiled map Editor" application and exporting a custom-made level design as a CSV file, the group created methods to import CSV data to display the world blocks, and apply custom attributes to each one of them (images, hitboxes, placement etc). To import the CSV data into the game the team used the BufferedReader class from Java.io. The CSV file implementation also allows players to import their own unique level designs into the game. This feature allows players to unleash their creativity and expand the game's content beyond the default level, fostering community engagement and expanding level design possibilities.

LADDER CLIMBING/MOVING FLOATING BLOCKS

The initial idea was to incorporate vines and ladders so the main character could reach higher areas within the world map. This feature was successfully implemented in the early stages of the game development and was fully functional. However, it was quite challenging to incorporate it with other coding mechanics when four coding styles were added into final project. Despite that setback, other gameplay mechanics were added in the later stages of the game to replace climbing, such as vertical and horizontal moving platforms, which is referred to as "floating blocks". This allowed the character to reach higher areas and added parkour capabilities, making the game world more interactive with the player as well.

TEAM CONTRIBUTIONS

This section outlines the contributions made by each team member.

CONTRIBUTIONS

For the beginning of Happy's Adventures, Luiz developed a template so the team could add, improve, or customise future game mechanics into the code. The template featured basic collision, basic player controls, character movement, and block creation.

MECHANICS AND CODING TYPES	TEAM MEMBERS
Floating/Moving Screen	Mishke and Willow
Character movement and Collisions	Everyone
Candy, fire, hearts interactions	Luiz, Lisinda
Scoring System	Everyone
Animation and Sprites Code	Luiz, Mischke, Lisinda
Enemies and interactions	Mishke, Willow
Friendlies and safe zone interactions	Mishke, Willow
MECHANICS AND CODING TYPES	
Planning and brainstorming	Everyone
CSV / World Map Design and Layout	Lisinda, Luiz
Menu Design	Lisinda, Luiz

GRAPHICS AND AUDIO	
Sprites and image creations	Luiz (Original creations, designed and created by Luiz)
Sound effects and background music	Luiz (Background music and sound effects were downloaded from PixaBay. All files are either royalty free or creative common.)
TESTING/DEBUGGING AND QUALITY CHECKING	
Reporting bugs/glitches, troubleshooting and fixing issues, optimising and improving the code	Everyone

FINAL THOUGHTS

LIMITATIONS AND FUTURE POSSIBILITIES

Despite the team's ambitions and initial plans, time constraints inherent to the game programming course limited the inclusion of certain features in the game. Ladder climbing, multiplayer functionality, additional levels, and enemies with specialized abilities were among the planned features that had to be set aside to meet project deadlines. Nonetheless, these ideas showcase the team's excitement for expanding the game's elements and learning new coding techniques in java.

FLEXIBILITY AND ADAPTABILITY

The team had to be flexible in adapting to unexpected challenges such as eventual bugs, mechanics that didn't work and extra features that were not fully resolved. As a result, the team had to adjust the game development accordingly. This required agile decision-making, problem-solving, and a willingness to explore alternative solutions.

LEARNING AND APPRECIATING GAME DEVELOPMENT

The development of Happy Adventures provided the Good Gamers team with valuable insights into the intricacies of game development. The project underscored the importance of collaboration, code consistency and version control, communication, and effective time management in achieving project goals within tight deadlines.

EXTERNAL LINKS:

Miro brainstorm Page:
https://miro.com/app/board/uXjVMLPWOyE=/

GitHub Page
https://github.com/urd-n-Skuld/Happy-Adventure-Game