Table of Contents

Name	*Contribution*	*Justification*
Cai Mengran	100%	Report & Flow Charts
Dai Luqiao	100%	Coding
Ding Yi	100%	Presentation slides & Report
Hu Xuanzheng	100%	Coding & Report
Jonathan Then	100%	Coding

All members have presented equal contributions to this project.

Algorithm Design

Flow Chart

The top-level flow chart is illustrated as in Figure 1. The inner illustration of the flying mode and the jumping mode are presented in the Appendix respectively.

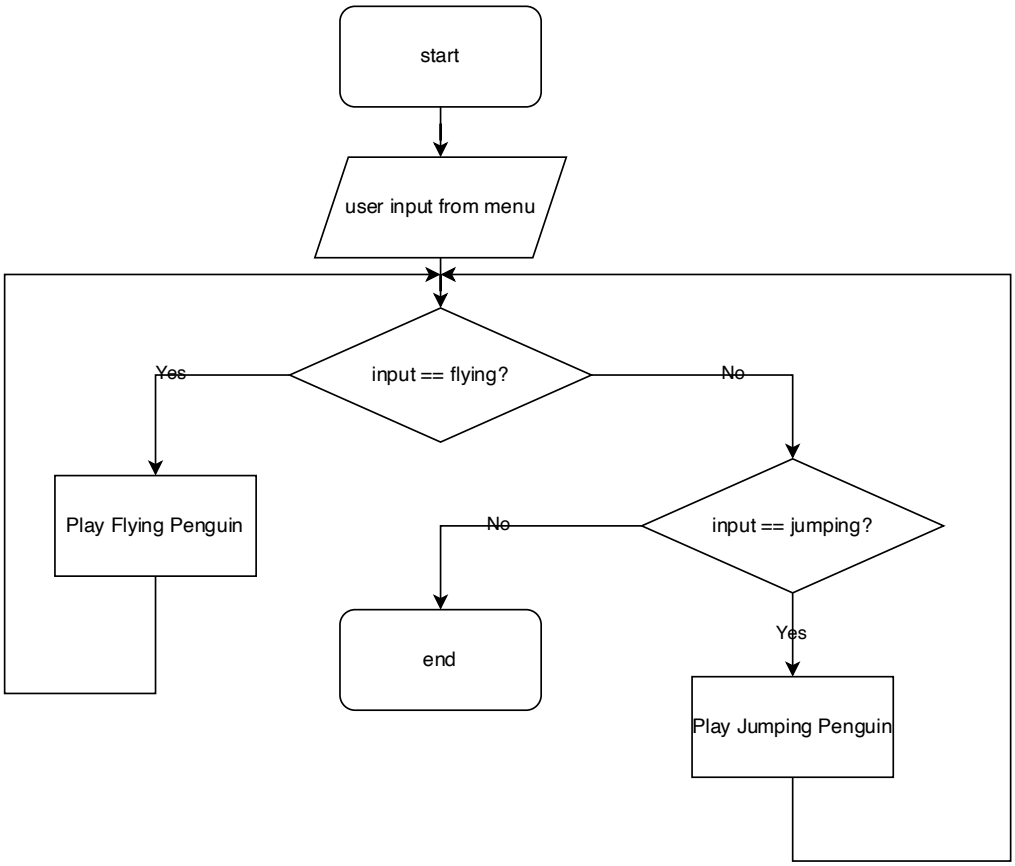


Figure 1. Flow Chart for Menu

User defined function description

The *main* game function includes the overall structure of the game. This session includes the key presses, penguin movement, obstacle movement, collision function and collision immunity, and scoring and health.

In the key pressing session, users' choices on either key down or boxes clicked will be captured. The game will be terminated if users press Escape or click on "QUIT"; and the game will proceed to the respective mode based on which mode they select. During the game, Space or Up keys will work as controlling the penguin's movement.

In the penguin movement session, different modes present different movements. In the flying mode, the penguin will be assigned an initial non-zero velocity, which allows it to not fall directly into the ground, and this value will be replaced by another velocity during flying. The new velocity takes into consideration the relative pipe moving velocity and gravity, which simulates the process of bypassing the pipes from both up and down. In the jumping mode, the penguin slips along the ground since beginning, and users are to press the key at the exact time spot to ensure that the penguin jumps over the obstacles. However, the common idea in both modes is that the penguin remains unchanged horizontally.

In the obstacle movement session, they are set to move leftward in both modes. In the flying mode, the "safe zone" between a pair of pipes varies each time in the vertical sense and its absolute value is fixed as one third of the flying area. This is achieved by setting up a default position initially for one pipe and adding up a random range for the pipe height adjustment. Another pipe will vary in conjunction to its pair.

In the collision session, two issues are addressed. First, collision immunity is defined as the time difference between the last collision and the current time spot. If the difference is less than our set up value (each mode has its customised value), collision immunity will work and protect users from losing health (1 heart at a time). Second, the distance between the penguin and the obstacles will be computed. If the penguin crashes the obstacles and the collision immune is False, the scoring and health function will work.

In the scoring and health session, the algorithm is straight forward. If the penguin survives in the previous sessions, the score will go up for 1, and the health remains unchanged; if the penguin fails in the collision session, score will remain unchanged, and health will go down for 1. Once health reaches zero, the game is over and will return to the initial selection step.

The following two functions are defined in both modes.

The main purpose of the *game_start* function is to start the game. Primarily, the images will be loaded, including the background, ice floor, penguin images, and so forth. A loop will be gone through to check the events. If the event type is QUIT or the user clicks on the ESCAPE key, the game will quit. If the user clicks on the SPACE or UP key, the game will start. And then refresh the screen and set the frame per second.

The main purpose of the *game_over* function gives a visualisation of the end. The font is called, and the text content including "Game Over", the final score received, and the instruction asking about restarting will be printed on the screen.

The following two functions are defined in the flying penguin mode only.

The main purpose of the *createPipe* function is to create the pipes that the penguin will encounter during the game. Firstly, the height of the flying area is defined by subtracting the ice floor thickness from the total height. Then an initial height is assigned to the pipe and later use the randrange function to generate the random height of pipes. Based on the height, the coordinates of the upper and down pipes will be determined and assigned to the list named "pipes". Then the list "pipes" will be returned.

The main purpose of the *isGameOver* function is to determine in which circumstances the game will end. One scenario is when the penguin reaches the sea level, another is when the penguin hits the obstacles, the game will end, considering the value of health goes to zero simultaneously. The algorithm is to get the upperpipe/downpipe height and then compare the height of the penguin and the height of the pipe.

Programme Testing

If a user enters an invalid input, the *handleInvalidKey* function will run. The algorithm will point out that a wrong press has happened and wait for the user's next valid input, which should be "ESC", "SPACE", or "UP".

Reflection

Difficulties encountered and solution

We have encountered several difficulties and the one of the biggest was that we did not have ideal pictures for the visualisation module. We retrieved photos from the internet, and it was hard to find the perfect ones that would fit into our games. The solution to this problem is that we took the picture from other similar games, and then modified the colour and appearance of the picture through the Internet.

The second difficulty that we encountered was due to the source of our pictures. Since the pictures were retrieved from the Internet, we could only change their colour and design but could not conduct any further customisations. When we were building up the collision part, the actual visualisation went beyond our theoretical model. The solution was to partially edit the size of the image model and reduce the actual collision model. The modification process allows the penguin to present a relatively more natural colliding effect.

Knowledge learnt from this course

It has been a fruitful journey to take this course. The major achievement has been working collaboratively in developing The Penguin. We have learnt to develop a strong mindset in facing up this coding assignment, especially under the condition that some other fellow groups consist of around three coding experts. However, leveraging each one's strength and bearing the teamwork spirit has enabled our group to get the assignment done, and systematically making it more mature. We would like to extend our sincere appreciation to the professor Dr. Zhang, the TA Dr. Chen, and all our fellow classmates that help with mutual peer learning.

Further improvement suggestion

First, the collision design of the game could be optimised. It is proposed to either have the images drawn by us or develop the collision model with irregular volume. For example, in the flying mode, the pipes can have different diameters; in the jumping mode, the houses and the

ice mountains can have different shapes. Besides, currently there is a doubt brought up by examiners. When the penguin loses one health value, sometimes it does not look like having collided into the obstacle. Optimisation regarding the collision functions codes and the corresponding visualisation can be carried out later.

Second, the user interface layout could be improved. The current menu window is written with the pygame package, which is appropriate but shows less attractiveness. Further research can be conducted from this perspective.

Third, the speeding function can be upgraded. In current versions, when the penguin passes a certain accumulated score, the movement speed of the obstacles will increase at a glimpse. However, it tends to be user-unfriendly and hence, a gradual acceleration of in-game obstacles is expected to be implemented.

Fourth, the visualisation can be optimised. For example, the penguin could flutter its wings during flying, or the penguin would roll its eyes once it hits an obstacle, and a little ghost-like penguin in shaded colour would rise from the penguin's body once it has lost all its health values.

References

Collision Detection in PyGame - GeeksforGeeks. (n.d.). GeeksforGeeks. https://www.geeksforgeeks.org/collision-detection-in-pygame/

Collision Detection in PyGame - GeeksforGeeks. (n.d.). GeeksforGeeks. https://www.geeksforgeeks.org/collision-detection-in-pygame/

Flappy Bird In Python Pygame With Source Code - Page 7 Of 8 - CopyAssignment. (n.d.). CopyAssignment. https://copyassignment.com/flappy-bird-in-python-pygame-with-source-code/7/

How to make Flappy Bird Game in Pygame? - GeeksforGeeks. (n.d.). GeeksforGeeks. https://www.geeksforgeeks.org/how-to-make-flappy-bird-game-in-pygame/

Images

Background (Image by kjpargeter on Freepik)

Free Photo | 3d winter snowy landscape. (n.d.). Freepik. https://www.freepik.com/free-photo/3d-winter-snowy-landscape_3640526.htm#query=ice%20background&position=5&from_view=search

Ice Floor (Image by upklyak on Freepik)

Free Vector | Game grounds with texture of ice, water and lava. (n.d.). Freepik. https://www.freepik.com/free-vector/game-grounds-with-texture-ice-water-lava_21329661.htm#query=ice%20floor&position=8&from_view=search

Penguin (Image by brgfx on Freepik)

Free Vector | Flying penguin animal cartoon sticker. (n.d.). Freepik. https://www.freepik.com/free-vector/flying-penguin-animal-cartoon-sticker_20500107.htm#page=2&query=penguin&position=0&from_view=search

Instructions

Press SPACE BAR to continue | Pixel Art Maker. (n.d.). Pixel Art Maker. http://pixelartmaker.com/art/112d2ed1d2d2316

Snow Cabin (Winter icons created by Freepik - Flaticon)

Lodge free icons designed by Freepik. (2022, September 16). Flaticon. https://www.flaticon.com/free-icon/lodge_1240067

Iceberg – Pointed (Freepik)

Free Vector | Set of different drawn icebergs. (n.d.). Freepik. https://www.freepik.com/free-vector/set-different-drawn-icebergs_9683376.htm#query=icebergs&position=6&from_view=search

Iceberg – Flat (Image by upklyak on Freepik)

Free Vector | Ice game buttons set. (n.d.). Freepik. https://www.freepik.com/free-vector/ice-game-buttons-set_12876458.htm#page=2&query=iceberg&position=29&from_view=search

Heart

Heart Pixel Art. (n.d.). Free PNG Download - Transparent PNG Images Free Download - SUBPNG / PNGFLY. https://www.subpng.com/png-srqdy0/

Shield

Energy Shield Png - Energy Shield Transparent PNG Image | Transparent PNG Free Download on SeekPNG. (n.d.). SeekPNG.com. https://www.seekpng.com/ipng/u2q8a9i1r5r5e6a9_energy-shield-png-energy-shield-transparent/

Sounds

Bird Flap Sound

BirdFlap2.wav by AgentDD. (n.d.). Freesound. https://freesound.org/people/AgentDD/sounds/246224/

Death Sound

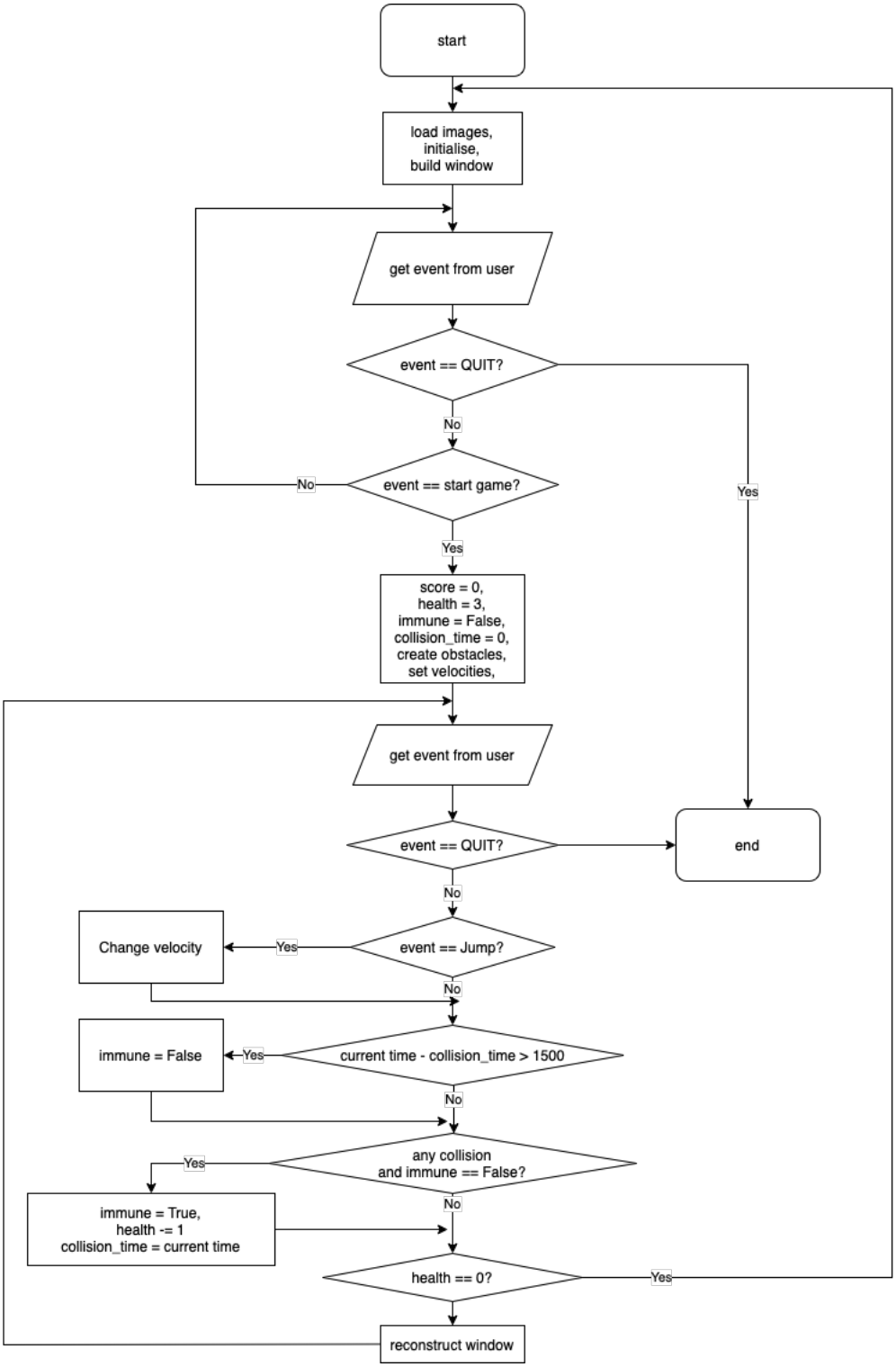
Dino Game in Python with Source Code (n.d.). https://download.code-projects.org/details/5a0d2fba-3abf-44fa-8a03-c4bd97cc3f78

Hurt Sound

Menu Tick.wav by LorenzoTheGreat. (n.d.). Freesound. https://freesound.org/people/LorenzoTheGreat/sounds/417792/

Appendices

Flow Chart for jumping penguin



Flow Chart for flying penguin