Project Flip Jump

Group 17

Mikael Ahlberg
Daniel Ericsson
Axel Stenkula
Johannes Svensson
Fredrik Vretblad

Functional requirements

High priority – Shall be included

- The user shall be able to play the game until the in-game character falls to the bottom of the screen, from the characters perspective.
- The screen shall move upwards (characters point of view) in a certain speed and the character needs to keep up with the pace or he/she will lose the game.
- The system shall be able to keep track of the high score list so the players can compare the results with their previous efforts.
- A tutorial text shall be displayed in order to provide help to the user.
- The user shall be able to choose an appropriate level of difficulty.
- The character shall be able to jump through the blocks on his way up, but he shall not be able to fall through them on his way down.
- The player will receive certain amount of points for every block he/she passes. That certain amount of points is based on the current speed and the current difficulty level.
- One second before the screen flips an indication of the flip direction will be shown on screen in the form of an arrow. The screen can only flip 90 degrees at a time.
- The system shall make the in-game character jump when the user presses the spacebar.
- The system shall move the in-game character to the right/left (seen from the characters point of view) when the user presses the left and right arrow keys.
- The screen speed shall increase if a higher difficulty level is chosen.
- To navigate in the menu system, you use the arrow keys (up and down), and to select an option you have to press Enter.
- After a random amount time the screen shall flip and the player needs to continue its journey in the new direction.

Medium priority – Should be included

- The screen size shall change when the user selects fullscreen or window mode.
- When the in-game character jumps outside of the screen (from his view: left or right), the character will then reappear on the other side of the screen at the same height.
- At each speed level there shall be three special items the player can collect. This item will provide the player with the ability of making a jump during a previous jump. The player can at a maximum only carry five of these items without using them.
- There shall be visual and audible indication when the speed level increase.

Low priority – Might be included

- The appearance of sound shall be optional through the menu system.
- The system shall play a special sound effect when the highest score on the list is beaten.

Non-functional requirements

Usability requirement
The user shall be able to learn how to play the game in one minute by reading the tutorial text.

Performance requirement
The game shall at most take five seconds to start after the user has read the tutorial text and pressed enter.

Space requirement
The game shall at maximum take 20MB, small enough to send between friends.

Portability requirement
The game shall be able to work on a set of different platforms due to Java's platform independence, at least on OSX, Windows and Linux

Use Cases

UC1: Start the game and change settings

Primary actor: Player

Stakeholders and interests: Player: Wants to start the game without having to wait too long for it to load.

Preconditions: The game is on the computer. The player has a computer with support for OpenGL and has Java runtime version 1.5 or later installed.

Success Guarantee (postconditions): The game has started, the user was able to change the settings and the game was able to run on his computer without any trouble.

Main Success Scenario (or Basic Flow):

1. The player starts the game by clicking on the game executable.
2. The player is greeted by a screen with a few options; Start game, Options, High Score, Quit.
3. The player selects Options by pressing the Down-key once and then Enter.
4. The player is greeted by a screen with three options; Fullscreen, Mute, Back.
5. The player selects Fullscreen and the window will go to fullscreen mode.
6. The player selects Back and returns to 2.
7. The player selects Start game.
8. The player is greeted by a screen with three options; Easy, Normal or Hard.
9. The player selects Easy.
10. The player is greeted by a screen with a tutorial text.
11. The player presses Enter to start the game.
12. The game starts.

Extensions:

5.
    a. The player selects Mute and the music stops and no sounds will be played in-game.
9.
    a. The player selects Normal.
    b. The player selects Hard.

Special requirements: -

Technology and data variations list: -

Frequency of occurrence: At least once every game session.

Open issues: -

UC2: Primitive game functions

Primary actor: Player

Stakeholders and interests: Player: Wants to play the game and survive as long as possible.

Preconditions: The player has started the game as in UC1.

Success Guarantee (postconditions): The character makes its way upwards without falling down outside the screen.

Main Success Scenario (or Basic Flow):

1. The character stands on the ground.
2. The player presses the left arrow key.
3. The character moves to the left of the screen.
4. The character moves too far to the left.
5. The character is immediately transported to the right side of the screen.
6. The player presses spacebar.
7. The character makes a jump to a block.
8. The player presses the right and the spacebar.
9. The characters makes a jump to the right.
10. There are no blocks at the right side of the screen.
11. The characters falls down to the bottom of the screen.
12. The game is over. See UC4.

Extensions: -

Special requirements: -

Technology and data variations list: -

Frequency of occurrence: Very often.

Open issues: -

UC3: Advanced game functions – Difficulty change

Primary actor: Player

Stakeholders and interests: Player: Wants its character to survive as long as possible.

Preconditions: The game is up and running and the player is currently playing the game. See UC2.

Success Guarantee (postconditions): The character survives the special events/functions that occur during the game and makes its way upwards without falling down outside the screen.

Main Success Scenario (or Basic Flow):

1. The character jumps to a new block and notices an arrow displayed on the screen.
2. The screen flips 90 degrees in the direction of the arrow a second later.
3. The player controlled character has now rotated 90 degrees and the in-game gravity has changed to accommodate the characters new perspective.
4. The character continues to jump for a couple of blocks.
5. The player hears a short sound clip that indicates that the screen will start chasing his character faster.
6. The player tries to increase his progress speed.

Extensions: -

Special requirements: -

Technology and data variations list: -

Frequency of occurrence: A frequent number of times during each gaming session.

Open issues: -

UC4: Advanced game functions - Items

Primary actor: Player

Stakeholders and interests: Player: Wants its character to survive as long as possible.

Preconditions: The game is up and running and the player is currently playing the game. See UC2.

Success Guarantee (postconditions): The character survives the special events/functions that occur during the game and makes its way upwards without falling down outside the screen.

Main Success Scenario (or Basic Flow):

1. The character jumps towards a block with an item on it.
2. The character lands on the item and notices how it disappears from the screen and reappears in the inventory list.
3. The player has now collected five of these items and the characters inventory list is now full.
4. The character tries to jump to a block higher up but misses.
5. The player presses the jump button once more while still in the air.
6. The character then gets an extra push upwards and lands on the previously missed block.
7. In the character inventory list, one used item gets removed.

Extensions: -

Special requirements: -

Technology and data variations list: -

Frequency of occurrence: A couple of times during each speed level.

Open issues: -

UC5: Game over

Primary actor: Player

Stakeholders and interests: Player: Wants to enter his/her name on the Highscore if he/she got enough points and then return to the menu.

Preconditions: The player has played the game and lost.

Success Guarantee (postconditions): The player has returned to the main menu and if the player got enough points, he/she has entered his/her name in the Highscore list.

Main Success Scenario (or Basic Flow):

1. The player lost the game.
2. The player beat at least one score on the Highscore.
3. The player is shown a screen where he/she is to enter his/her name for the Highscore list.
4. The player enter his/her name and presses Enter.
5. The player is shown a screen where his/her score and the top of the Highscore list is shown. A Back option is also shown.
6. The player selects Back and returns to the main menu.

Extensions:

2.
    a. The player didn't beat any score on the Highscore and goes directly to 5.

Special requirements: -

Technology and data variations list: Reads/writes the Highscore from/to a file.

Frequency of occurrence: Every game session.

Open issues: -

UC6: Show Highscore

Primary actor: Player

Stakeholders and interests: Player: Wants to view the Highscore without having to wait too long for it to load.
Preconditions: The game is already started.

Success Guarantee (postconditions): The player has seen the Highscore list.

Main Success Scenario (or Basic Flow):

1. The player is greeted by a screen with a few options; Start game, Options, Highscore, Quit.
2. The player selects Highscore by pressing the Down-key twice and then Enter.
3. The player is greeted by a screen with a list of Name, Block-level and Score and also the option Back.
4. The player selects Back and returns to 1.

Extensions: -

Special requirements: -

Technology and data variations list: Reads the Highscore list from a file.

Frequency of occurrence: Low.

Open issues: -

UC7: Quit
Primary actor: Player

Stakeholders and interests: Player: Wants to exit the game without having to wait for it to close down.

Preconditions: The game is running and the player is located in the main menu.

Success Guarantee (postconditions): The game was successfully closed.

Main Success Scenario (or Basic Flow):
1. The player navigates down to the Exit option in the main menu.
2. The player presses Enter and the game exits.

Extensions: -

Special requirements: -

Technology and data variations list: -

Frequency of occurrence: Every time.

Open issues: -