

Stranded Alien

Game Design Document

Contents

<i>Game Design</i>	2
Summary	2
Gameplay	2
<i>Technical</i>	3
Screens	3
Controls	4
<i>Level Design</i>	5
Themes	5
Game Flow.....	6
<i>Development</i>	7
Objects and variables	7
<i>Graphics</i>	15
Graphics - Serious.....	15
Graphics - Not so serious	19
<i>Sounds/Music</i>	24
Sounds	24

Game Design

Summary

Navigate ever moving platforms to collect points and boosts. Avoid falling to your death or getting caught by the ghost.

Gameplay

The character that the player controls is a spaceship that can't fly. It can hover, move left and right and jump. The goal of the game is collecting treasure that appears at random locations on the screen. To reach the treasure, the character must jump to and „walk“ on moving platforms. Some platforms move horizontally while others move vertically. The platforms interact; whenever a platform's path is blocked by another platform, it reverses it's direction. This makes the path of platforms a little less obvious. The character can only hover when on a platform or on top of the ghost, otherwise it feels the effect of gravity. Jumping is not possible while feeling the effects of gravity (falling) except when the mid-air jump boost is activated (will be explained shortly). Besides treasure, the character can collect boosts. There are three kinds of boost: jump boost (increases height of the jump), speed boost (increases speed of movement along the x axis) and mid-air jump boost (creates a single-use platform under the character that allows him to jump mid-air). The former two last for a period of time. The duration of the bonus is displayed on the top side of the player for the jump boost and on the right side for the speed. The latter lasts until used. A semi-transparent platform is drawn under the player to indicate that the boost has been activated. Jumping from anywhere you could normally jump from (without the bonus) while the boost is activated will not use up the platform created by the mid-air jump boost.

There are three ways to die: get caught by the ghost (jumping on top of the ghost is perfectly safe, though, just be careful getting off), fall to the bottom of the screen or get too high over the top of the screen (carried by the ghost, you can't jump out of bounds from any platform, even with the jump boost). When the character is out of the screen, an arrow appears to follow his position on the x axis and the distance along the y axis from the top of the screen is written besides the arrow. If the character surpasses the top of the screen by a full height of the screen, he dies.

There are three difficulty settings, each with a different starting speed (and skin) of the ghosts. The ghost gets faster with every 2000 points you collect at any difficulty.

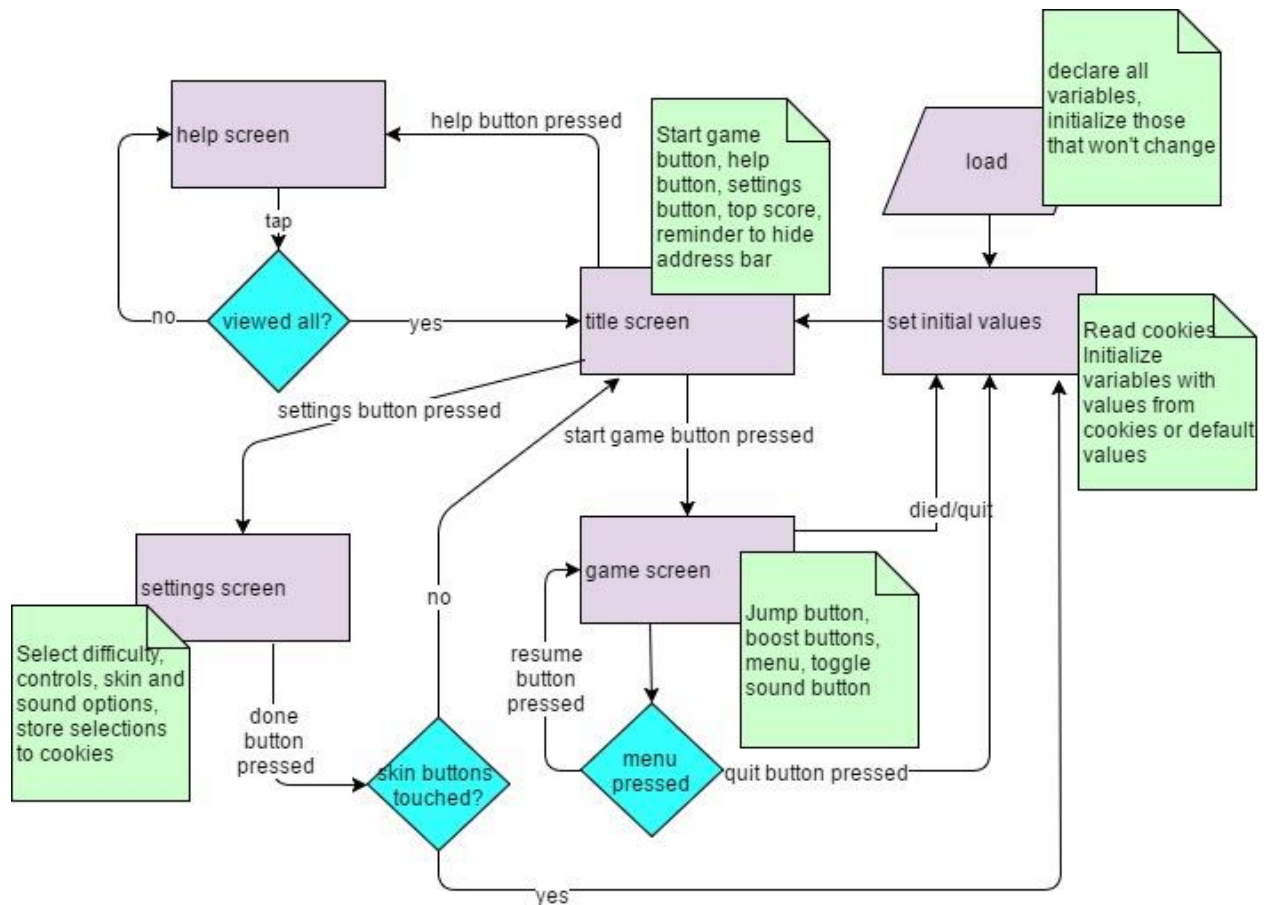
The settings screen is not available while playing. Exiting the game screen will result in loss of progress. However, the toggle sound button is available within the game screen and a simple menu allows the player to pause and resume or quit the current game.

All treasure is positioned in such a way that it is accessible without the use of any boost. Although not crucial, boosts can make collecting a treasure easier, can play a big role in „outsmarting“ the ghost and can even save the character from a potentially lethal situation.

Technical

Screens

1. Title Screen
 - Start game button
 - Help button
 - Settings button
 - Top score display
 - Message (reminder to hide the address bar=
2. Help Screen*
 - screen one: controls
 - screen two: skin
 - screen three: laout
 - screen four: goals
 - screen five: miscellaneous
 - *tap anywhere selects next help screen and closes help after the last screen
3. Settings Screen
 - difficulty buttons (easy, medium, hard)
 - sound buttons (on, off)
 - control buttons (gyro, arrows)
 - skin buttons (serious, not so serious)
4. Game screen
 - jump button
 - toggle sound button
 - menu button (expands to resume and quit buttons)
 - boost buttons (jump, speed, mid-air jump)
 - arrow buttons (if arrows controls selected)



Controls

Depending on personal preferences, the player can choose to control the character by tilting the device or by pressing on screen arrows. The arrows are only displayed when the option is chosen from the settings menu. Jumping and activating boosts is accomplished by touching respective buttons on screen. The boost buttons are moved up to accommodate the arrows when the option is selected.

Level Design

Themes

There is a choice of graphics, available from the settings menu. The „not so serious skin“ uses hand-drawn (mock) graphics but the game is still fully functional. The game is only playable in landscape orientation, a message warning the user of that fact appears in portrait orientation. Both skins are space themed featuring platforms moving through space with distant planets in the background.

Game Flow

The character starts the game falling but is in no danger since he is over a platform. This was chosen so that the characters position would be more noticable at start (since it's moving). There is no predetermined flow, the player will notice the position of the treasure and start making his way towards it in the dynamical environment while avoiding being caught by the ghost or falling to his death. The ghost starts at the bottom right corner of the screen and always moves towards the character.

The player can pause the game at any time.

Development

Objects and variables

Objects:

1. Igrac - contains parameters pertaining to player position and motion, animation (sprites) and bonuses

Parameters:

X : position coordinate x (number)

Y : position coordinate y (number)

Dx : displacement along x axis (number)

Dy : displacement along y axis (number)

visina: player height (number)

sirina: player width (number)

skok: height of jump (number)

sirinaSpritesheet: width of the player spritesheet (number)

visinaSpritesheet: height of the player spritesheet (number)

brojRedakaSpritesheet: number of rows in the player spritesheet (number)

brojStupacaSpritesheet: number of columns in the player spritesheet (number)

redakZaGore: index of row of spritesheet to be used for animating movement up (number)

redakZaLijevo: index of row of spritesheet to be used for animating movement left (number)

redakZaDesno : index of row of spritesheet to be used for animating movement right (number)

redakZaRavno : index of row of spritesheet to be used for animating no movement (number)

get sirinaSprite(): calculates the width of a sprite (width/number of columns) (function, returns number)

get visinaSprite(): calculates the height of a sprite (height/number of rows) (function, returns number)

aktivniKadarSprite: index of the active sprite frame (number)

kadarX : x coordinate of the sprite frame (number)

kadarY: y coordinate of the sprite frame (number)

tickCount: number of game frames that have used the current sprite frame (number)

ticksPerFrame: number of game frames to use the same sprite frame (number)

idiLijevo: is a control to move the character left active (boolean)

idiDesno: is a control to move the character right active (boolean)

mrtav: is the character dead (boolean)

2. Zvezdica - contains parameters pertaining to player position and motion

Parameters:

prozirnost: transparency (alpha) for treasure (number)
prozirnostD: amount of alteration of prozirnost (number)
X: position coordinate x (number)
Y: position coordinate y (number)
ispisiSkupljeneBodove: counter for the „+100“ message, used both for duration of the message and modifying the position of the message (animating it) (number)
xZaBodove: position coordinate x for the „+100“ message
yZaBodove: position coordinate y for the „+100“ message

3. Boost - contains parameters pertaining to boost type, location and duration

Parameters:

boostZaSkokTrajeJos: remaining duration of the jump boost (number)
dostupanBoostZaSkok: number of jump boosts available (number)
boostZaBrzinuTrajeJos: remaining duration of the speed boost (number)
dostupanBoostZaBrzinu: number of speed boosts available (number)
mozeSkokIzNiceg: is the mid-air jump bonus activated (boolean)
dostupanSkokIzNiceg: number of mid-air jump boosts available (number)
bonusAktivan: is a bonus available for collecting (boolean)
ispisiKojiBoost: type of boost collected (string)
ispisiGdje: remaining duration of the message stating the type of boost collected
xZaIspisiKojiBoost: position coordinate x for the boost type message
yZaIspisiKojiBoost: position coordinate y for the boost type message

4. Duh - contains parameters pertaining to ghost position, motion and messages

Parameters:

X : position coordinate x (number)
Y : position coordinate y (number)
Dx : displacement along x axis (number)
Dy : displacement along y axis (number)
visina: player height (number)
sirina: player width (number)
timerZaPorukuDaUbrzava: remaining duration of the message stating that the ghost speed has increased (number)

Variables:

1. skin: used to select the graphics pack (string)
2. paused: flag used to pause/unpause game (boolean)

3. brojackadrova: used to count frames between appearances of a boost (number)
4. boost: image for the boost (HTMLImageElement)
5. canvas: everything is drawn here (HTMLCanvasElement)
6. sirinaCanvasa: width of the canvas (number)
7. visinaCanvasa: height of the canvas (number)
8. ctx: context of the canvas (CanvasRenderingContext2D)
9. wrapper: the div used in landscape mode (HTMLDivElement)
10. requestAnimationFrame: a variable containing the appropriate prefixed version of the rAF function (window.requestAnimationFrame || window.webkitRequestAnimationFrame || window.mozRequestAnimationFrame || window.oRequestAnimationFrame || window.msRequestAnimationFrame)(function)
11. zvuk: flag used for turning sound on/off (boolean)
12. gyro: flag used for choosing controls (gyroscope or arrows) (boolean)
13. brzinaDuha: the starting speed of the ghost (number)
14. muzikaZaMenu: the sound to be played at the menu screen (HTMLAudioElement)
15. muzikaZaIgru: the sound to be played during gameplay (HTMLAudioElement)
16. maliSkok: the sound to be played when jumping without the jump boost active (HTMLAudioElement)
17. velikiSkok: the sound to be played when jumping with the jump boost active (HTMLAudioElement)
18. boostSound: the sound to be played when a boost is collected (HTMLAudioElement)
19. zaKraj: the sound to be played when character dies (HTMLAudioElement)
20. skupi: the sound to be played when treasure is collected (mostly useless because most browsers won't allow sounds that are generated without the user taking explicit action like pressing a button) (HTMLAudioElement)
21. igrac: the image used for the character spritesheet (HTMLImageElement)
22. up: the image used to mark the character's x coordinate when the character is out of the screen (HTMLImageElement)
23. duhDesno: image for the ghost facing right (HTMLImageElement)
24. duhLijevo: image for the ghost facing left (HTMLImageElement)
25. gumbZaSkok: button used for jumping (HTMLButtonElement)
26. gumbZaBoostSkok: button used for activating the jump boost (HTMLButtonElement)
27. gumbZaBoostBrzina: button used for activating the speed boost (HTMLButtonElement)
28. gumbZaSkokIzNiceg: button used for activating the mid-air platform boost (HTMLButtonElement)
29. gumbZaZvuk: button used for turning sound on/off from within the game screen (HTMLButtonElement)
30. hideA: paragraph used for reminding the player to hide the address bar (HTMLParagraphElement)
31. gumbPocni: button used for starting the game (HTMLButtonElement)
32. gumbPomoc: button used for opening the help screen (HTMLButtonElement)
33. gumbPostavke: button used for opening the settings screen (HTMLButtonElement)
34. gumbOdpauziraj: button used for unpausing the game from the game screen (HTMLButtonElement)

35. gumbOdustani: button used for quitting the current game(HTMLButtonElement)
36. gumbMenu: button used for pausing the game and creating buttons gumbOdpauziraj and gumbOdustani (HTMLButtonElement)
37. pozadine: images used for the blurring effect at the start of the game and each time the ghost's speed increases (array of HTMLImageElement)
38. efektZaPozadinu: counter used for indexing the array of images in pozadine while creating the effect (number)
39. platforme: image for the platforms (HTMLImageElements)
40. visineZaPlatforme: heights for each platform (array of numbers)
41. sirineZaPlatforme: widths for each platform (array of numbers)
42. ikseviZaPlatforme: x coordinates for each platform (array of numbers)
43. ipsiloniZaPlatforme: y coordinates for each platform (array of numbers)
44. okomiti: indices fo platforms with vertical paths (array of numbers)
45. okomitiSmjerovi: directions of the vertically moving platforms (-1 = up, 1 = down) (array of numbers)
46. vodoravni: indices fo platforms with horizontal paths (array of numbers)
47. vodoravniSmjerovi: directions of the horizontally moving platforms (-1 = left, 1 = right) (array of numbers)
48. zvjezdica: image used for treasure (HTMLImageElement)
49. rezultat: used to keep score (number)
50. topScore: a button used to display the high score at the menu screen (HTMLButtonElement)
51. topScoreValue: the value of the top score (string)
52. kontrolaLijevo: left arrow button when controls are set to arrows (HTMLButtonElement)
53. kontrolaDesno: right arrow button when controls are set to arrows (HTMLButtonElement)
54. svaPomoc: images for the help screen (all the images) (array of HTMLImageElements)
55. diraoSkin: flag used to see if buttons for selecting the skin have been touched (if they have, the page reloads once the player closes the settings screen so the skin changes can take effect) (boolean)
56. gumbDiffEasy: select easy difficulty from the settings screen (HTMLButtonElement)
57. gumbDiffMedium: select medium difficulty from the settings screen (HTMLButtonElement)
58. gumbDiffHard: select hard difficulty from the settings screen (HTMLButtonElement)
59. gumbZvukOn: turns sound on from the settings screen (HTMLButtonElement)
60. gumbZvukOff: turns sound off from the settings screen (HTMLButtonElement)
61. gumbGyroOn: selects gyroscope controls(HTMLButtonElement)
62. gumbGyroOff: selects arrows controls (HTMLButtonElement)
63. gumbSkinNormal: selects „serious“ skin (HTMLButtonElement)
64. gumbSkinNotSoNormal: selects „not so serious“ skin (HTMLButtonElement)
65. gumbZatvoriPostavke: closes the settings menu, removes the buttons, sets cookies and reloads the page if skins have been changed (HTMLButtonElement)
66. nagib: acceleration of the device along the y axis including gravity (number)

Functions:

dodijeliDimenzije() – checks screen dimensions, sets the canvas width and the sirinaCanvasa variable to the value of the larger of the two dimensions and sets the canvas height and the visinaCanvasa variable to the value of the smaller of the two.

postaviNaPocetneVrijednosti() – sets the values of all the variables that change values during the game to their default values. This includes image url's (they change when different skins are selected). Where applicable, it checks cookies for the default value. This function is called on load and every time the game is restarted.

obradiKretanje(event) – interprets the devicemotion event and sets the movement of the character accordingly.

pokreniIgru() – prepares for game start

- checks the selected control method.
 - If „controls: gyro“ is selected (gyro=true), adds the devicemotion event listener to the window with the obradiKretanje function.
 - Else, creates arrow buttons and adds event listeners to them.
- After it creates the controls, it sets the initial speed of the ghost according to the selected difficulty level and sets the ghost image.
- It then removes the menu buttons by calling makniGumbeZaMeni() function and
- starts the game by calling the pokreni() function.

pokreni() – checks whether images have loaded. If they have, it sets the game buttons by calling the postaviGumbeZaIgru() function and starts the game loop (azuriraj function). Else, it writes a message and calls itself in 0.2 seconds.

animirajZvezdicu() – prepares the image used for animating the treasure for drawing (sets opacity)

jeZvezdicaSkupljena() – checks whether the treasure has been collected. If it has, it increases the score, sets the coordinates and duration of a message to be drawn and selects coordinates for the now position of treasure. It tries to play a sound but this is blocked by most browsers.

jeBonusSkupljen() – checks whether the boost has been collected. If it has, it marks the boost as collected, sets the coordinates and duration of a message to be drawn and chooses the type of boost at random. It increases the number of available boosts of the selected type, updates the number in the text of the appropriate button and sets the text of the message.

animirajIgraca() – creates the player sprite to be drawn in the next draw cycle

naPodlozi() – checks whether the character is located on a platform

naLiftu() – checks whether the character is located on a vertically moving platform

skoroNaLiftu() – checks whether the character is located one point above a vertically moving platform.

naDuhu() – checks whether the character is located on top of the ghost

nacrtaj() – handles the draw cycle.

- Clears the canvas.

- Checks whether it is time to animate the background (it needs to animate when the game starts and when the ghost increases in speed(level up)).
- Checks whether the character has died.
 - If it has, draws the end game image and game over message and restarts the game after 2 seconds.
 - If the character is alive,
 - it draws the platforms.
 - It then checks whether the character is in the inside the screen.
 - If it is, it draws the character. If the mid-air jump boost is active, it draws a small semi-transparent platform under the character.
 - If the character is not inside the screen, it draws an arrow at the top of the screen showing where the character is on the x axis and the number representing the distance of the character from the top of the screen along the y axis.
 - If the speed or jump boosts are active, shows the remaining duration of the boost to the right and above the character, respectively.
 - Draws the ghost. Draws the „ghost speed increased“ if appropriate
 - Draws the treasure
 - Draws the boost if available
 - Draws the score
 - Displays messages about treasure or boost being collected if appropriate

miciVodoravnePlatforme() – moves platforms horizontally. For each platform, checks the current direction. If the platform can move in that direction (no colision), it moves the platform (changes x coordinate). If it can't, it reverses the direction.

miciOkomitePlatforme() – moves platforms vertically. For each platform, checks the current direction. If the platform can move in that direction (no colision), it moves the platform (changes x coordinate). If it can't, it reverses the direction.

pomaknilgraca() – calculates the position of che character. The y coordinate is calculated based on whether he is falling, standing on a stationary or horizontally moving platform, a vertically moving platform, 1 point above a vertically moving platform (important for correct functioning of the vertically moving platforms), on top of the ghost and whether he is jumping (and whether the jump boost is activated). Tha x coordinate is calculated based on player controls and whether the speed boost is active.

pomakniDuha() – moves the ghost towards the player. Sets the appropriate image for the ghost (movng left or right)

provjere(naDuhuSam) – checks whether the character has died. It recieves the „on top of the ghost“ flag from pomaknilgraca so it doesnt have to recalculate it. The character has died if he reached the bottom of the screen, was caught by the ghost (but is safe on top of the ghost), or went a hight of the canvas over the top of the screen ($y < -visinaCanvasa$)

azuriraj() – handles the update cycle

- hides boost buttons if no boosts of that type are available

- reduces all duration timers
- moves platforms by calling `miciVodoravnePlatforme()` and `miciOkomitePlatforme()`
- moves player by calling `pomakniIgraca()`
- moves ghost by calling `pomakniDuha()`
- checks whether the character is dead by calling `provjere()`
 - If the character is dead, stops music, checks to see if high score has been achieved and stores it in a cookie if it has
 - If the character is alive
 - chooses player sprite by calling `animirajIgraca()`
 - checks whether a treasure has been collected by calling `jeZvezdicaSkupljena()`
 - animates the treasure by calling `animirajZvezdicu()`
 - increases the speed of the ghost with every 2000 points collected
 - makes a boost available every 5000 frames
 - draws the updated state by calling `nacrtaj()`
 - starts the next frame by calling `requestAnimationFrame(azuriraj)`

pomoc() – handles the help screen. If there are images left to display, displays image by calling `nacrtajPomoc()`. If there aren't, removes the event listener and exits help (clears screen and sets menu buttons)

nacrtajPomoc() – If the image has not loaded, it displays a message and calls itself in 0.2 seconds. If it has, it displays the image.

otvoriPostavke() – handles the settings screen. It generates buttons for easy/medium/hard difficulty, sound on/off, controls gyro/arrows and skin serious/not so serious. Adds event listeners which raise the appropriate flag for each button pressed and change colors of buttons by calling `obojajPozadinu()`. Generates the „done“ button which, when touched, saves new values of flags to cookies and removes all of the settings buttons and resets the game if the skin has been changed (so the new images are loaded).

obojajPozadinu() – sets the colors of settings buttons. The active (selected) value is green, others are red.

postaviGumbeZaIgru() – makes all game screen buttons visible (jump, boost jump, boost speed, mid-air jump boost, sound and menu)

makniGumbeZaIgru() – makes all game screen buttons invisible (jump, boost jump, boost speed, mid-air jump boost, sound and menu)

postaviGumbeZaMeni() – makes all menu screen buttons visible (start, help, settings, top score, hideA)

makniGumbeZaMeni() – makes all menu screen buttons invisible (start, help, settings, top score, hideA)

postaviGumb(gumb, x, y) – sets the position, left and top css attributes of an element

setCookie() – sets cookie

getCookie() – reads cookie

gumbZaSkok.ontouchstart – event handler for the jump button. Checks whether the character can jump (he can jump if he's on a platform, on top of the ghost or has the mid-air

jump boost active). If he can, sets the height of the jump based on whether or not the jump boost is active

gumbZaBoostSkok.ontouchstart – event handler for the boost jump button. If the jump boost is available and not already active, activates it (sets the duration to max), updates the available amount of available jump boosts, updates the button text and plays a sound.

gumbZaBoostBrzina.ontouchstart – event handler for the boost speed button. If the speed boost is available and not already active, activates it (sets the duration to max), updates the available amount of available speed boosts, updates the button text and plays a sound.

Boost.mozeSkokIzNiceg – event handler for the mid-air jump boost button. If the mid-air jump boost is available and not already active, activates it (sets the duration to max), updates the available amount of available mid-air jump boosts, updates the button text and plays a sound.

gumbZaZvuk.ontouchstart – event handler for the sound button. Toggles sound, updates cookie, changes image

gumbPocni.ontouchstart – event handler for the start game button. Sets player to not dead, if sound is enabled stops menu music and starts game music, starts the game by calling `pokrenilgru()`.

gumbPomoc.ontouchstart – event handler for the help button. Opens the help screen by calling `pomoc()`.

gumbMenu.ontouchstart – event handler for the menu button. Pauses the game and creates resume and quit buttons.

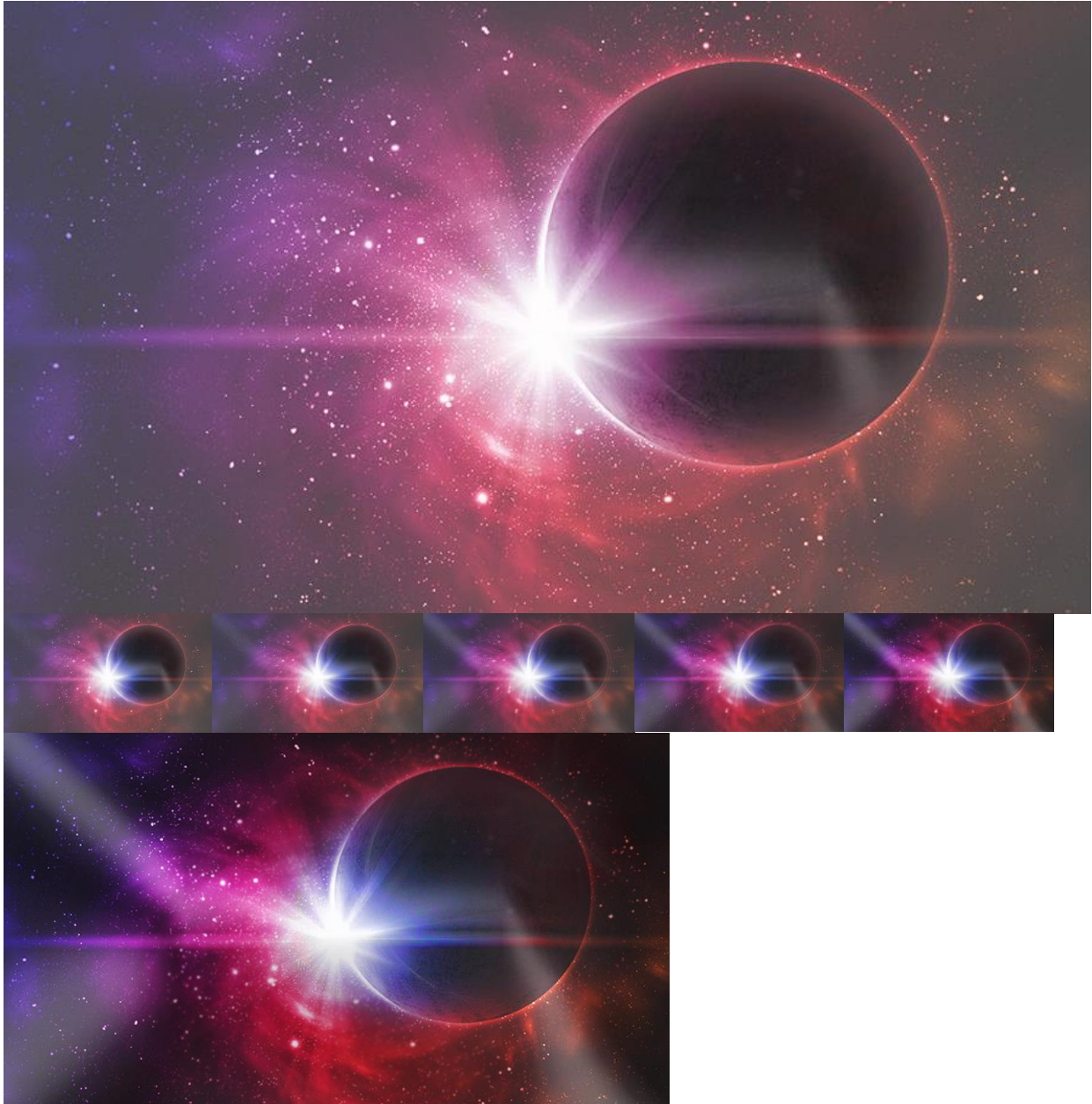
gumbOdpauziraj.ontouchstart – event handler for the resume button. Resumes the game (sets paused to false and calls `azuriraj()`) and removes the resume and quit buttons

gumbOdustani.ontouchstart – event handler for the quit button. Quits to main menu (resets all values to default) by calling `postaviNaPocetneVrijednosti()`.

Graphics

Graphics - Serious

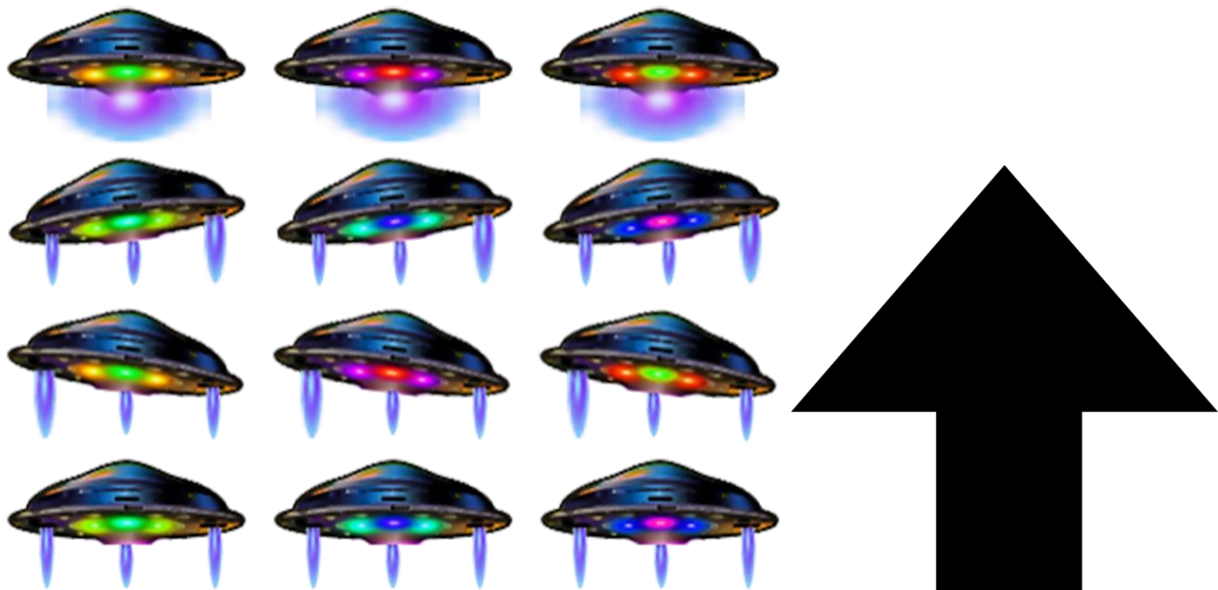
Backgrounds:



Buttons:




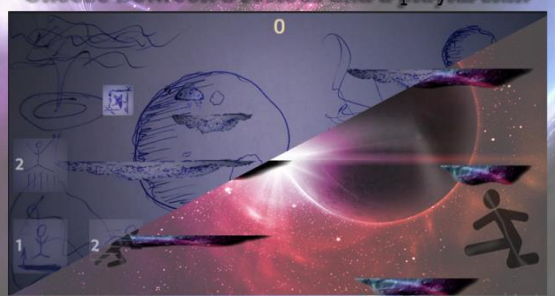


Character:



Ghost:

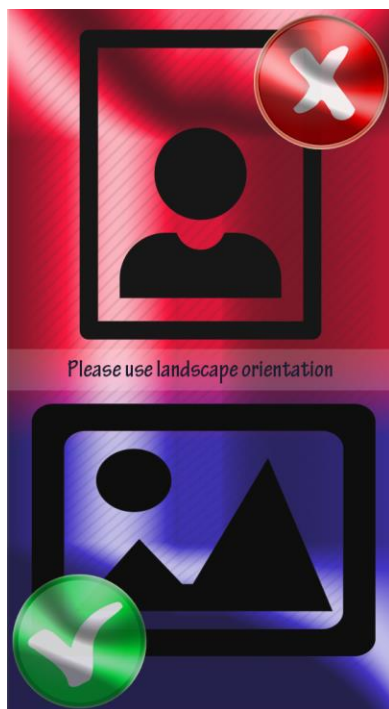


Help:

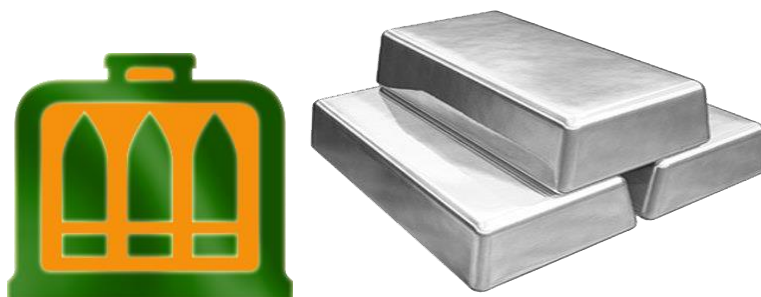
SELECT CONTROLS IN SETTINGS	CHOOSE SKIN
 <p>GYRO: move the character by turning your device</p> <p>ARROWS: move the character by touching the arrows</p>	 <p>Choose between a serious and a playful skin</p>
LAYOUT	GOALS
 <p>Platform you can stand on and jump from it</p> <p>Ghost will eat you but not if you're on it</p> <p>You can safely stand on and jump the ghost</p> <p>Jump bonus increases jump altitude</p> <p>Platform bonus creates a single-use platform under player</p> <p>Boost Speed increases movement speed</p> <p>Sound on/off</p> <p>Jump button</p>	 <p>Collect points</p> <p>Collect boosts</p> <p>Don't die you die if you get caught by the ghost fall to the bottom of the screen or ride the ghost too far over the top</p>



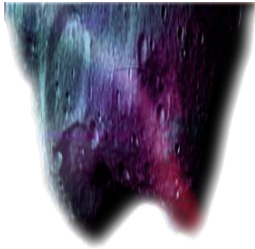
Portrait orientation:



Collectible:

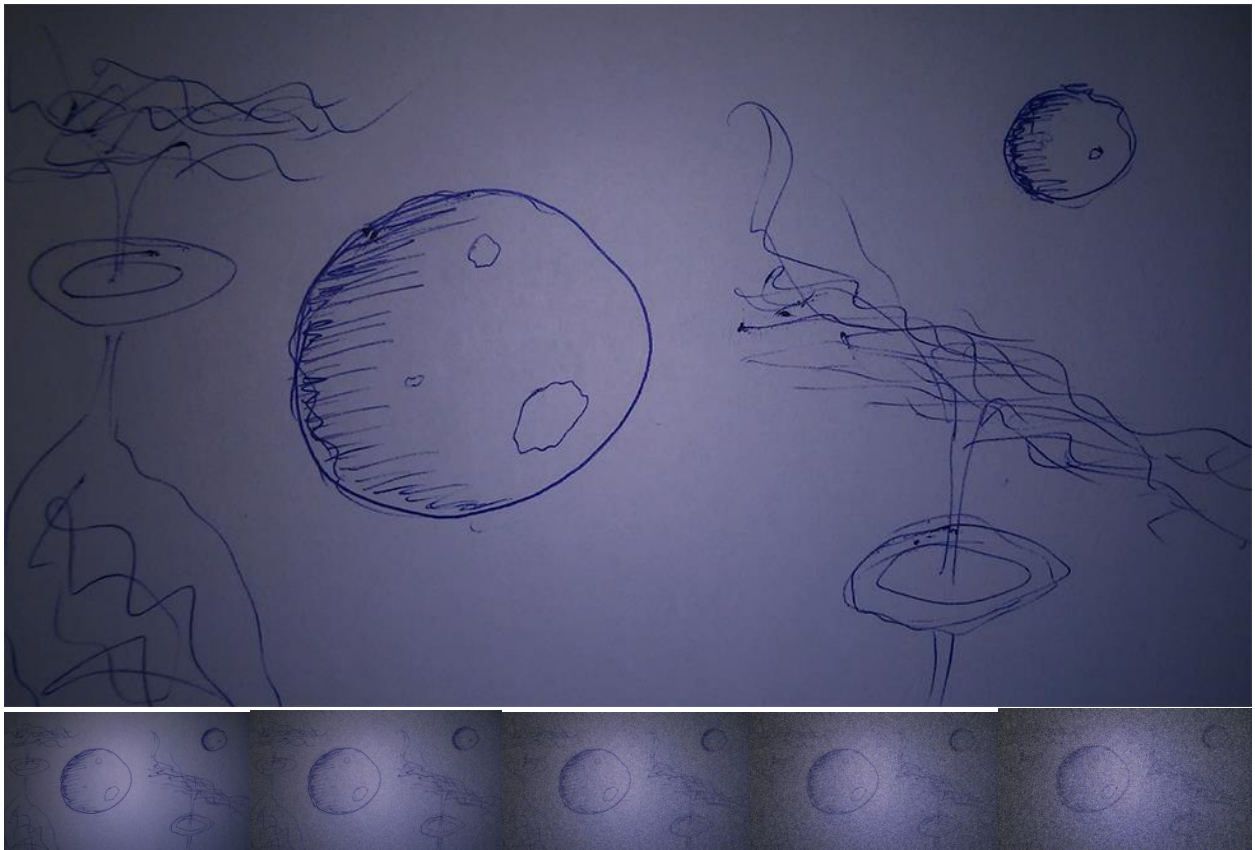


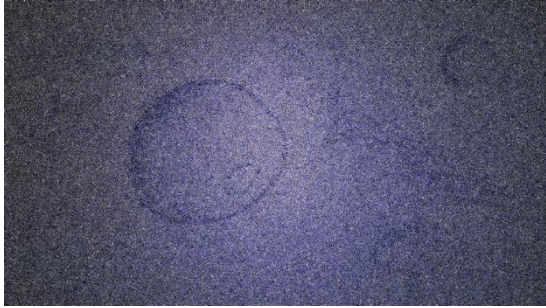
Platform:



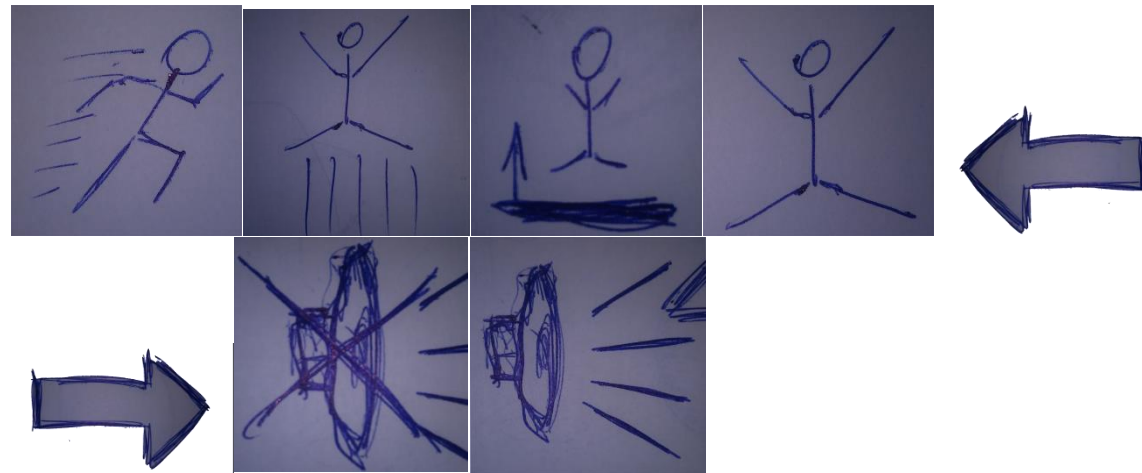
Graphics - Not so serious

Backgrounds:

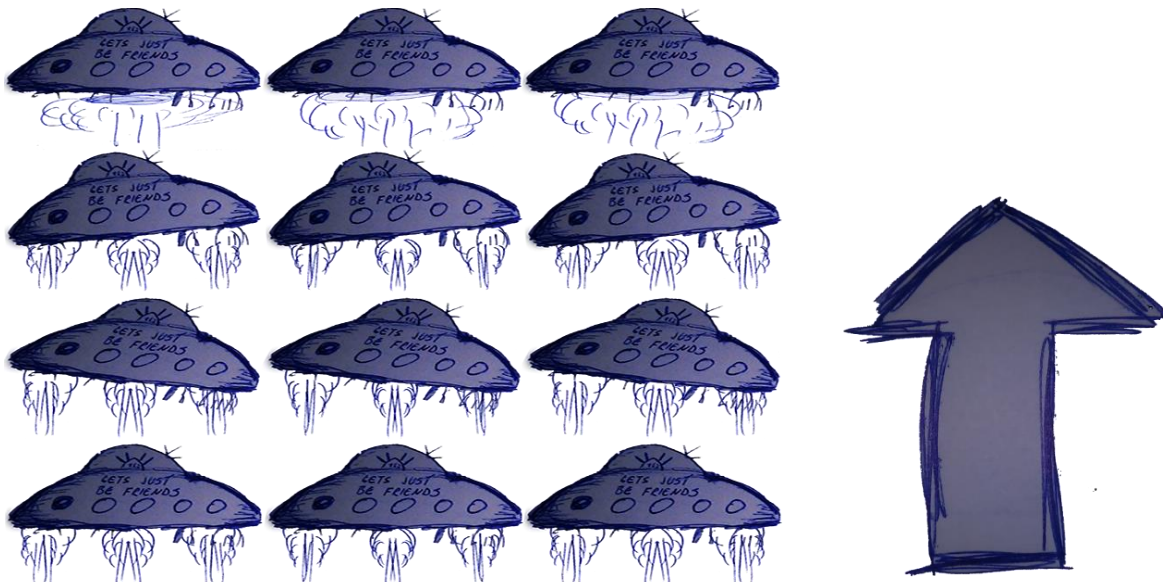




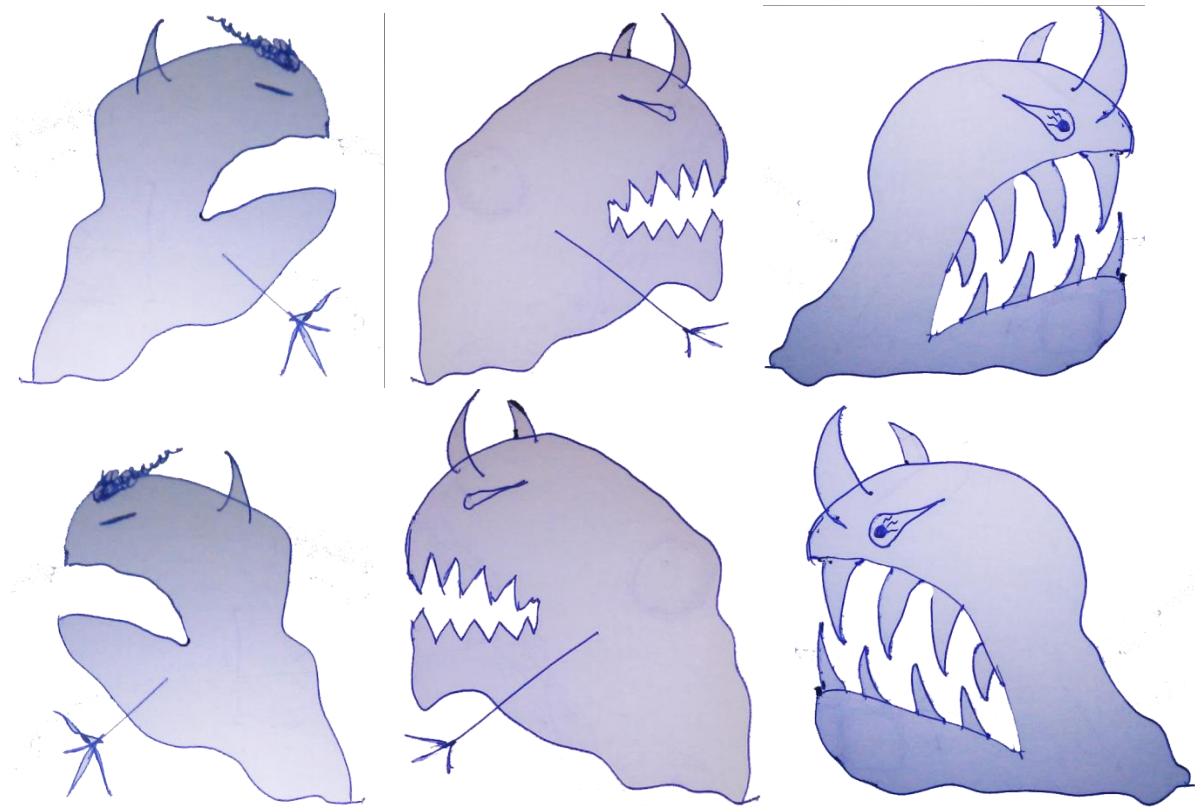
Buttons:



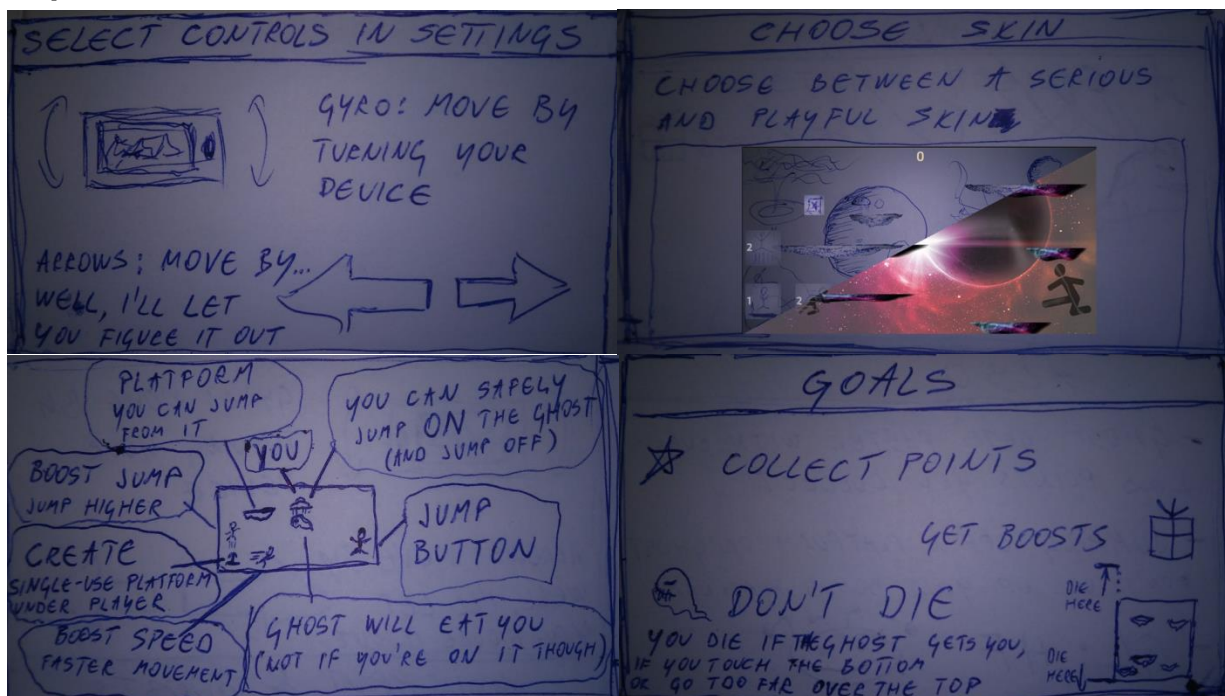
Character:

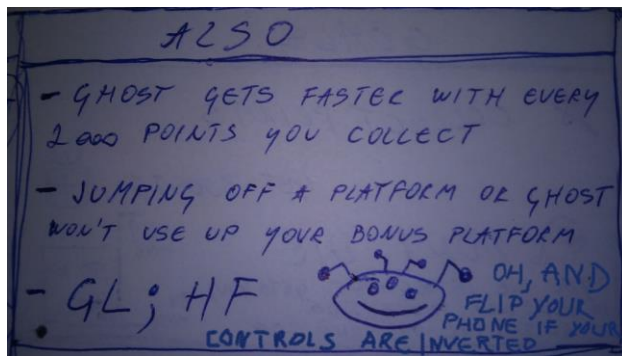


Ghost:

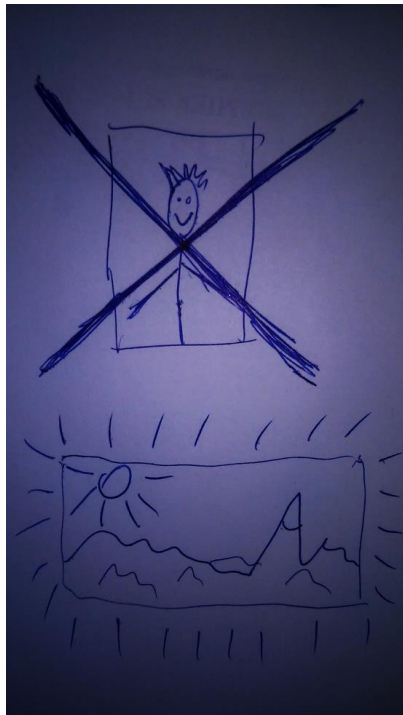


Help

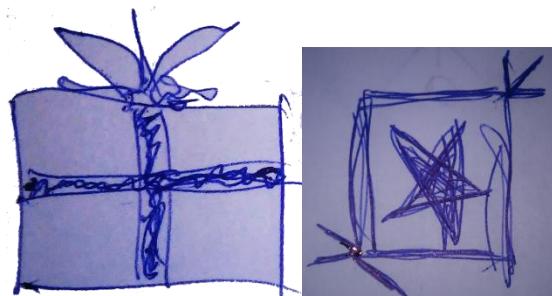




Portrait



Collectible



Platform



Sounds/Music

Sounds

Serious: notSoNormalSkin\zvukici\

boost.mp3

igra.mp3

maliSkok.mp3

menu.wav

skupi.mp3

velikiSkok.mp3

zaKraj.mp3

Not so serious: normalSkin\zvukici\

boost.mp3

igra.mp3

maliSkok.mp3

menu.wav

skupi.mp3

velikiSkok.mp3

zaKraj.mp3