Maths space

Technical Manual

Game idea/ description

For my game idea, I have chosen to go with the idea of a space shooting maths game. It involves a space ship that has to blast its enemies in order to progress to further levels and win. The enemies will have a maths question for example ‘6x8’ and in order to shoot the enemy, the player has to type the correct answer into the spaceship and the spaceship will shoot. As the enemies go by they get faster and faster, and if they crash into the player then they lose a life.  
Overall I will be creating a game where players shoot enemies by answering maths questions, but as the enemies come faster and faster; the maths questions get harder and harder!

Sales pitch

This game is aimed for kids and young teenagers and it is suitable because although it is fun, it has an educational impact on the players, allowing them to learn and remember maths questions that they will use for the future. As a game designer the main goal would be to get the players well addicted to this game, and so I will create the game and make it so enjoyable that they will all be hooked to it.

Storyboard

**Here is a simple spaceship, and in the window you are to type in your answers to allow you to shoot your enemies**

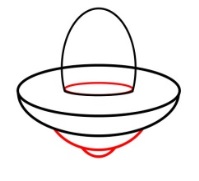
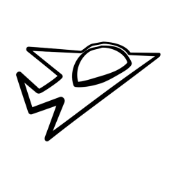
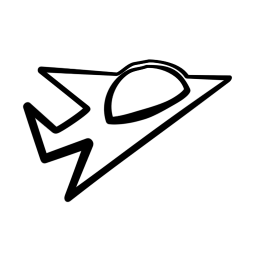
**When the enemy comes forward, in order to shoot you are to enter the answer into your spaceship**

**When you type in the correct answer (e.g. 6) the spaceship allows you to shoot**

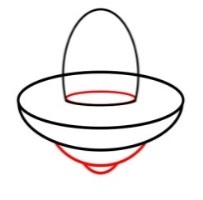
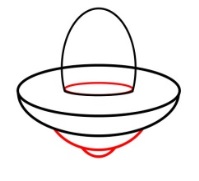
**Once you have shot your enemy will blow up/die and your score will increase**

**However, if you type in the wrong answer or waste time, the enemy will get closer and closer to you**

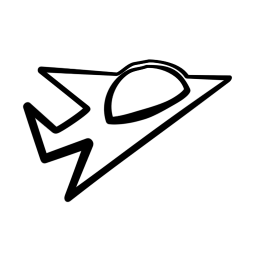
**If the enemy is close to you and manages to touch you, you will blow up and a life will be taken away and also lose points, and so a new enemy will appear**



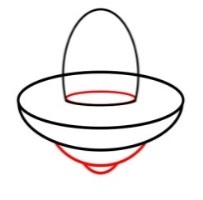
4+2



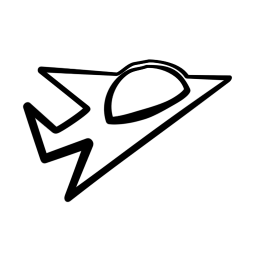
4+2



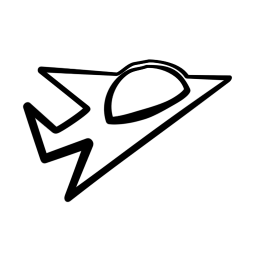
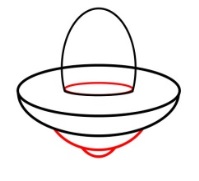
6



4+2



8



Lives

**Score**

10

**Score**

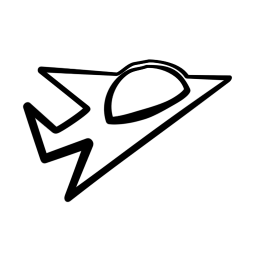
5

**Score**

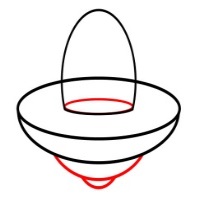
0

Gladys Nzita-Mak

Maths Space Game



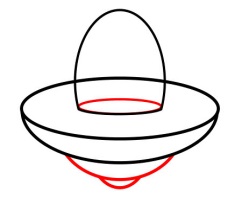
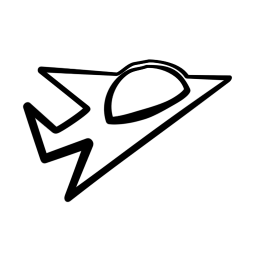
6



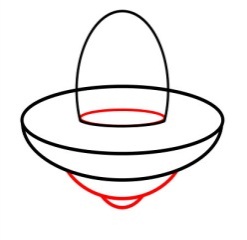
**In order to get to the next level, there will be a button available to press which will direct the user to the next level**

**Once the button is clicked the user will still have their remaining lives and score, and will have a new question to answer**

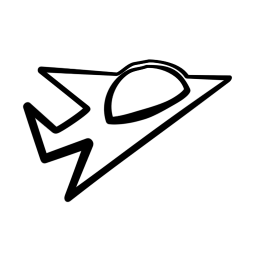
**Just like normal, the user will have to answer the questions correctly to score more points, and the questions will get harder and harder as the levels go on**



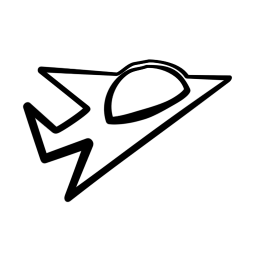
4+2



6x8



48



6

Lives

**Score**

10

**Score**

10

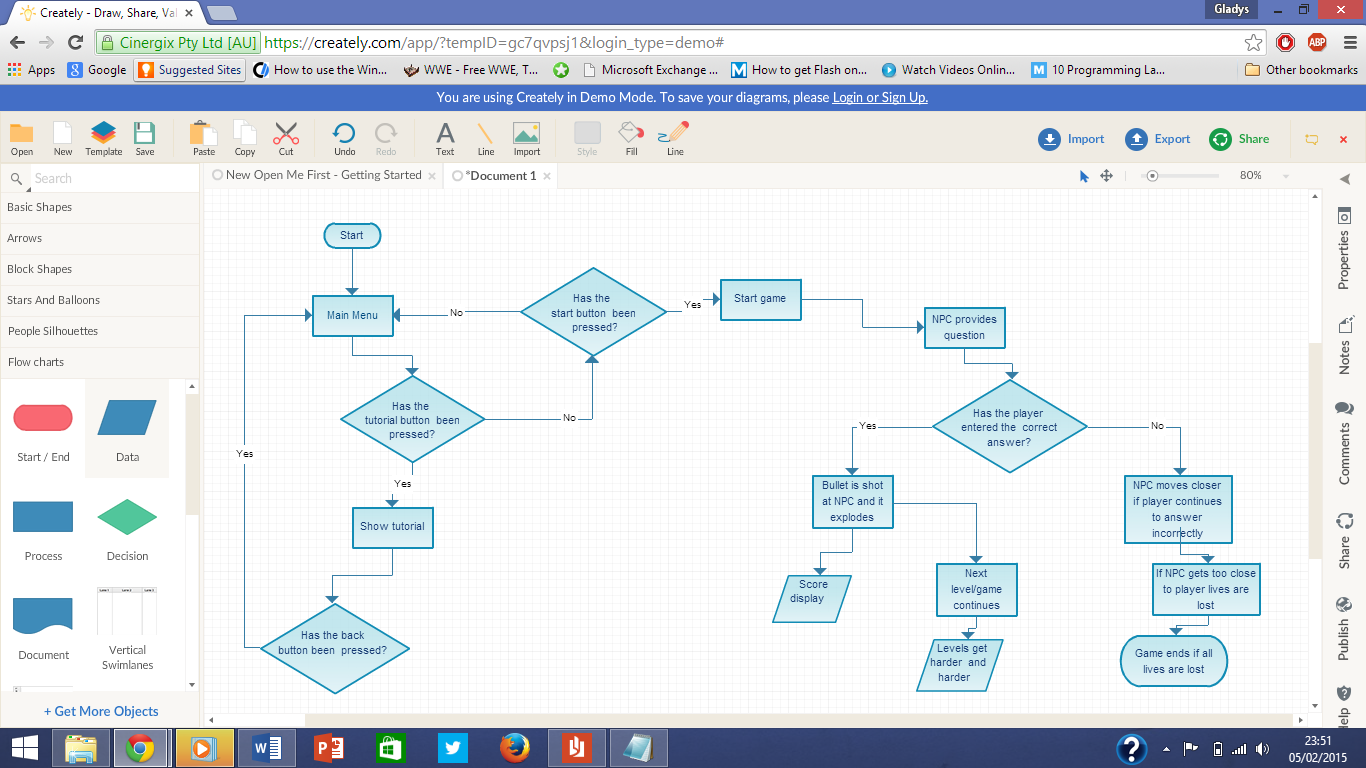
**Score**

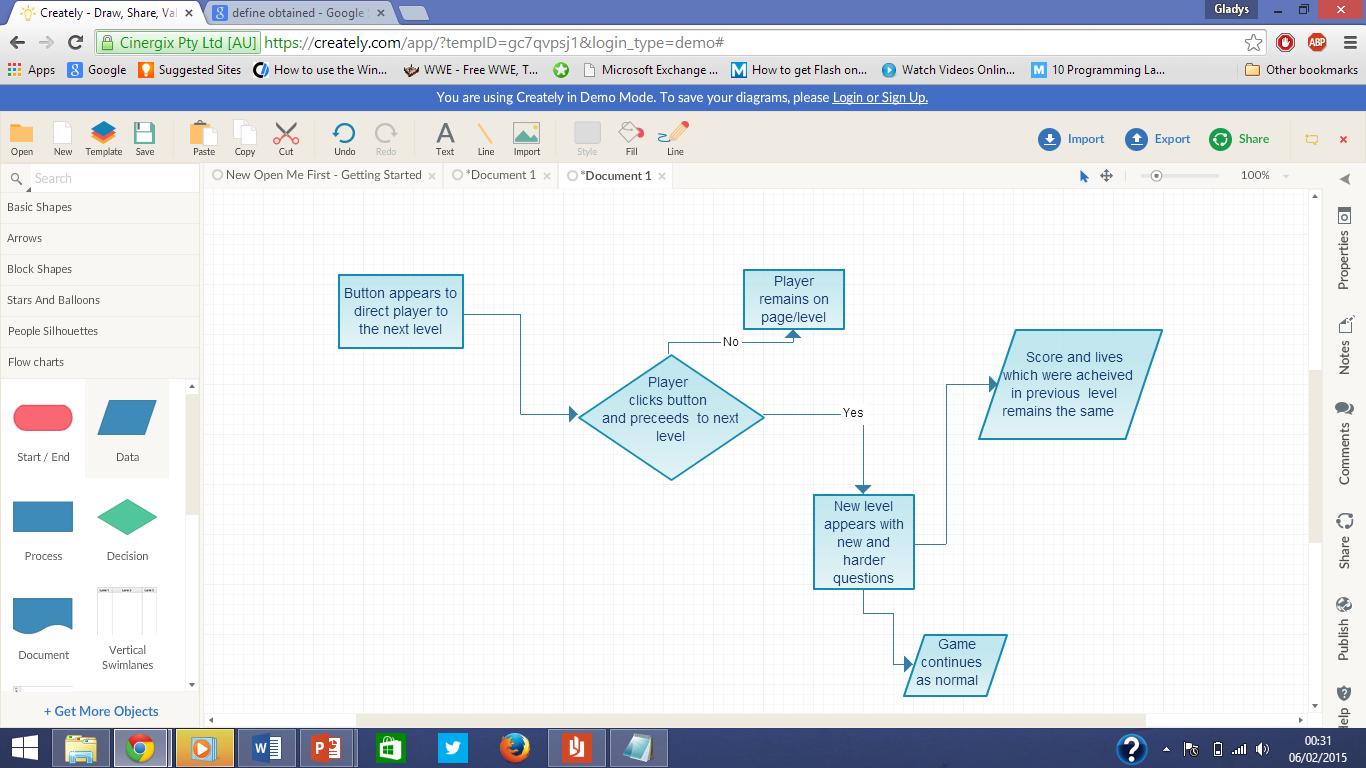
10

This is the storyboard design for the game, and also below is the flowdiagram.

Flow diagram

Level 1





Level 2

|  |  |  |
| --- | --- | --- |
| Frame 1 | Game codes listed for main frames & screenshots | stop() ;  startButton.addEventListener (MouseEvent.CLICK, playAnimation);  function playAnimation(evnt: MouseEvent){  gotoAndPlay (2);  }  stop();  Tutorial.addEventListener(MouseEvent.CLICK, toTutorial);  function toTutorial(e: MouseEvent){  gotoAndStop(5);  }  var myCurrentFrame :Number = 1;  var myScore: Number =10; |
| Frame 3 |  | stop();  MMenu.addEventListener(MouseEvent.CLICK, movetoMain1);  function movetoMain1(e: MouseEvent){  gotoAndStop(1);  }  MMenu.addEventListener(MouseEvent.CLICK, movetoMain1); |
| Frame 2 |  | stop()  Level1.addEventListener (MouseEvent.CLICK, startlevel);  function startlevel(evnt: MouseEvent){  gotoAndPlay (10);  }  Level2.addEventListener (MouseEvent.CLICK, secondlevel);  function secondlevel(evnt: MouseEvent){  gotoAndPlay (14);  }  MMMenu.addEventListener(MouseEvent.CLICK, movetoMain3);  function movetoMain3(e: MouseEvent){  gotoAndStop(1);  }  MMMenu.addEventListener(MouseEvent.CLICK, movetoMain3); |
| Frame 10 | import flash.events.Event;  stop ()  myCurrentFrame = currentFrame;  stage.addEventListener(KeyboardEvent.KEY\_DOWN,downHandler);  stage.focus = this;  //KEY\_UP //downHandler //upHandler  var bullet : Bullet = new Bullet();  var HScores: Array = new Array ();  HScores.push(20);  HScores.push(30);  HScores.push(40);  HScores.push(50);  HScores.push(60);  var i :int;  HScores.sort ();  for(i=0; i<5; i++){  trace(HScores[i]);  HScores[i] = 100;  }  function downHandler(e:KeyboardEvent){  }else if (bullet.hitTestObject(twenty1)){  stage.removeEventListener(KeyboardEvent.KEY\_DOWN,downHandler);  removeEventListener(Event.ENTER\_FRAME, flyBullet);  removeChild (bullet);  myScore = myScore -10;  scoreDisplay.text = String (myScore);  gotoAndStop(7);    }else if (bullet.hitTestObject(twenty2)){  stage.removeEventListener(KeyboardEvent.KEY\_DOWN,downHandler);  removeEventListener(Event.ENTER\_FRAME, flyBullet);  removeChild (bullet);  myScore = myScore -10;  scoreDisplay.text = String (myScore);  gotoAndStop(7); | switch (e.keyCode){  case 37:// left  shooter.x -=10;  break;  case 38: // up  shooter.y -=10;  break;  case 39: // right  shooter.x +=10;  break;  case 40: // down  shooter.y +=10;  break;  case 32: // spacebar  //shooter.x  if(bullet != null){  addChild(bullet);  bullet.x=shooter.x +30;//50;  bullet.y=shooter.y+20; //100;  //removeChild(bullet);  //bullet = null;  addEventListener(Event.ENTER\_FRAME, flyBullet);  }else {  bullet = new Bullet();  addChild(bullet);  bullet.x=shooter.x +30;//50;  bullet.y=shooter.y+20; //100;  addEventListener(Event.ENTER\_FRAME, flyBullet);  }  }  }  function flyBullet (evt: Event){  bullet.x = bullet.x +10;  //trace(bullet.x);  checkHits();    if(bullet.x > 550){  if(bullet.parent != null){  (bullet.parent).removeChild(bullet);  bullet = null;  removeEventListener(Event.ENTER\_FRAME, flyBullet);  }  }  function checkHits(){  if(bullet.hitTestObject(twenty4)){  stage.removeEventListener(KeyboardEvent.KEY\_DOWN,downHandler);  removeEventListener(Event.ENTER\_FRAME, flyBullet);  removeChild(bullet);  myScore = myScore +50;  scoreDisplay.text = String (myScore);  gotoAndStop(6);          }else if (bullet.hitTestObject(twenty3)){  stage.removeEventListener(KeyboardEvent.KEY\_DOWN,downHandler);  removeEventListener(Event.ENTER\_FRAME, flyBullet);  removeChild (bullet);  myScore = myScore -10;  scoreDisplay.text = String (myScore);  gotoAndStop(7);  }  }  }    backmenu.addEventListener(MouseEvent.CLICK, movetoMain);  function movetoMain(e: MouseEvent){  gotoAndStop(2);  } |
| Frame 6 |  | stop()  Nextq.addEventListener(MouseEvent.CLICK, movetoq2);  function movetoq2(e: MouseEvent){  gotoAndStop(myCurrentFrame + 1);  } |
| Frame 7 |  | stop()  goback.addEventListener(MouseEvent.CLICK, movetoq1);  function movetoq1(e: MouseEvent){  gotoAndStop(myCurrentFrame );  } |
| Frame 11, 12, 14, 15 &16 |  | stage.addEventListener(KeyboardEvent.KEY\_DOWN,downHandler);  stage.focus = this;  myCurrentFrame = currentFrame; |
| Frame 13 |  | stop ()  Level2.addEventListener (MouseEvent.CLICK, ndlevel);  function ndlevel(evnt: MouseEvent){  gotoAndPlay (14);  }  myCurrentFrame = currentFrame;  backmenu.addEventListener(MouseEvent.CLICK, moveMain);  function moveMain(e: MouseEvent){  gotoAndStop(2);  } |
| Frame 17 |  | backmenu.addEventListener(MouseEvent.CLICK, moveback);  function moveback(e: MouseEvent){  gotoAndStop(2);  } |

Testing the game

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tests number | Purpose of tests | input | Expectation | Actual outcome | Did it go well as expected? | Comments |
| **1** | Getting the game to load | Load up the game | For the game to load up when opened | The SWF filed loaded up well without freezing | Yes it went well as the game loaded up well |  |
| **2** | Check if the start button is working | Start button | Direct player straight to game | When the player clicks the button it directs them to the game immediately | Once it clicked it directed the player to the game – yes |  |
| **3** | Does the tutorial button work? | Tutorial (how to play) button | Takes the player to the tutorial page where they learn how to play | When clicked the player is taken to the tutorial page | Yes, when the button is clicked the player is directed to the tutorial page |  |
| **4** | To check if the back to menu button directs player to the menu screen | Back to menu button | Should direct the player back to the main menu when clicked | After being clicked it directs the player back to the start page | Once clicked it goes straight back to the main menu, so yes |  |
| **5** | If the ship is moving | Shooter | When the player uses the arrow keys the shooter that is to be controlled should move either up, down, left or right | As the arrow keys are being pressed, the shooter moves up, down, left or right as directed | Yes, using the arrow keys made the shooter move as expected |  |
| **6** | Testing whether the bullet that comes out of the shooter directs user to another frame when they hit the right or wrong answer | Hit test | If the bullet that comes out of the shooter hits either both answers, it directs the user to a right or wrong answer page | When the bullet has been shot and lands on either answer, it successfully goes to the frame it has been coded to go to | Everything went well as expected |  |
| **7** | For the next button to work | Next button | After the first question is answered correctly, the user would have to click “next” to get to the 2nd question | When the button is pressed it takes the player to the next question | Yes, as expected when the next button is clicked it directs the player to the next question |  |
| **8** | When the spacebar is hit, a bullet will shoot out of the shooter | Bullet | When the player presses the spacebar key, the bullet will fly out | The bullet appears when spacebar is hit | It went well as planned |  |
| **9** | On the levels menu, whichever level button the player presses, it shall direct the player to either one | Levels menu  Buttons leading to other levels | For all buttons to work; to direct players to the levels they select | When buttons are hit they move from the frame that they are on, to the frame of whichever level is selected | Yes it went as expected |  |