308338219

bennooo

A Brief description of how I implemented the game:

1. The alien formation -

The Aliens formation class contains list of aliens, each alien contains speed and a block with specific location so, the aliens in the list will appear in structure. Furthermore, the Aliens formation has method that will set the location of all the aliens contained in the list, each frame will reset the structure by new x value for each alien. So, the structure will move by the x value. Once the structure will get to the right\left limit the all structure will reset the y value. Each level the speed of all aliens inside the structure will be set to be faster by 10 percent. Case the player got hit by the ball or that the structure hit the shield, the speed value will be set to the speed of the level.

2. The shield –

The shield contains blocks, we initial the shield in specific structure. As block reacts, each time a ball will hit the shield one little block will be removed from the shield. The shield itself will be reset at the start of each level.

3. Shot by aliens –

I implemented the shot of aliens in the function "doOneFrame" – than I encounter some problem, the ball will be shot each frame as opposed to the instruction of the game. To deal with this problem I added member in the class "coolDownEnemy" the members in an int initialed to zero. When "doOneFrame" is executed automatically we decrease -dt from the " coolDownEnemy " and then I checked if the " coolDownEnemy " is below zero, create a "shoot" (ball), then initial the "cooldown" member to 0.5, that way we earn time between each shot, as the instructions of the game specify.

4. Shot by player –

I implemented the shot of player in the function "doOneFrame" – than I encounter some problem, the ball will be shot each frame as opposed to the instruction of the game. To deal with this problem I added member in the class "coolDownPlayer" the members in an int initialed to zero. When "doOneFrame" is executed automatically we decrease -dt from the " coolDownPlayer " and then I checked if the " coolDownPlayer " is below zero and the space keyboard is pressed, create a "shoot" (ball), then initial the "cooldown" member to 0.35. that way we earn time between each shot, as the instructions of the game specify.

New classes:

AlienCollection – The alien collection got a list of aliens in specific structure.

Alien – The alien contains a "block", and method which can set the block values.

Shield - The shield will contain structure of blocks, which will "defend" the player from getting hit from alien.

Rest of the class:

CountdownAnimation - This func will countdown whenever each game start's.
It will implement animation, so it can be drawn on given surface.

EndScreen - This is the last screen who's shown, when the game ends.

GameLevel - The game class initialize the game itself. It creates a screen,
and draw all the blocks, paddle and balls on the screen. Using two functions runs and initialize.
initialize in charge of creating all object and run in charge of starting the game.

HighScoreAnimation - This class is resposible of the high score animation.
It holds all the functions that supposed to screen, open and save scores.

KeyPressStoppableAnimation - This class is responsible to activate all the static screens that appears in the game, such as end screen, pause and high score table.

MenuAnimation - generic parameter.
This class is responsible of showing the menu of the game.
In the menu will be some options the player can choose from like
start a game, show high score table and exit the game.

PauseScreen – This class in charge of stopping the game.

DrawRectangle – This class will be in charge of drawing rectangle.

Line - This class will set a line. each line has start and end point. and each class got few functions he can do. measure the line length, return the middle point, of the line. Checking if it will intersect with other lines. Checking if other line is equal to him, and check if some point is on the specific line.

Point - This class will set a point. Each point got to values x's / y's. A point can use some functions, like, measure the distance from other line, checking if other point is equals to another. And return the values of the methods.

Rectangle - The rectangle gets the upper left point the width and the height,
and build the shape by those parameters. Can return his parameters.
And can find some intersection point between him and some line.

BallRemover - This class implements HitListener So it can "hear" notification from hitting objects. And remove balls that got out of the bounds.

BlockRemover - This class will be HitListener. It will need to be updated if was any hits. if there is. It should remove the block from the game.

ScoreTrackingListener - This class listen to the blocks removed and keep the score updated.

LevelIndicator - Indicates which level is it.

LiveIndicator - Indicates number of lives left.

ScoreIndicator - This interface can be drawn and use time passed func.
indicates the score of the player.

AnimationRunner - This class will be responsible to run the game.

Level - This class will hold all the information about the level, which was created in the "factory".

Ball - This class will create balls. Ball will be created by center value, radius, color and boundaries that ball can move into. as well as a frame that can limit him as well.

Block - The block will be collidable, so each ball who will collide with him will bounce back. furthermore, the ball block will implement the sprite interface, so he can be drawn on the screen.

Paddle - The paddle will be collidable, so each ball who will collide with him will bounce back. Furthermore, the block will implement the sprite interface, so he can be drawn on the screen. The ball can move and will return balls the hits him different from regular block.

Ass7Game – contains the main and run the game.

CollisionInfo - The class contains information about the collision with some collidable object. The collision point and the object that another object collided with.

ColorBackground – This class responsible of the background.

Counter – This class is in charge of counting.

ExitTask – This class responsible of exit the game.

GameEnviorment - This class got member of all the collidable objects in the game, and can find the collision with some line with all the collidables inside this member list.

GameFlow - This class is in charge of the game flow.
It will be responsible that each level will start after the other.

HighScoreTable – this class is responsible of the high score table.

ImageBackground - This class responsible of the background.

RunGameTask – This class will run the game task.

ScoreInfo – This class will hold info about the high score .

ShowHighScoreTask - This class is a task that in some point will run and show the high score.

SpriteCollection - The class contains list of sprite, adding all the sprite into list.

Velocity - This class will set the ball velocity.
each and every ball got velocity of his own. and the velocity is one of the method each and every ball has. we can change and decide which way the ball will move to, using the balls velocity values.

-

InterFaces:

Animation - This interface responsible for the animation of the game.

Collidable - This interface can return the collision object.
and can be hit by object.

GamebackGround - This class responsible of backgrounds.

HitListener - Inteface of the class who needs to "hear" hit events.

HitNotifier - This func will notify all the hit listeners that was a hit.

LevelInformation – information of the level.

Menu – Generic interface contains all the menu details

Sprite – This interface can be drawn on given surgace.

Task – run some generic task.