

Follow the [SE456 Final Project 2019.pdf](#) for details.

Fill this form out with Adobe Reader (DO NOT USE anything else)
Save As [Final Submission YOURNAME.pdf](#) place in perforce folder
11:59 pm CST (midnight) Due Date: 18 March 2020

Final Project: Space Invaders

Student Information

Integrity Policy: All university integrity and class syllabus policies have been followed. I have neither given, nor received, nor have I tolerated others' use of unauthorized aid.

I understand and followed these policies: Yes No

Name:

Date:

Submission details

Code Path:
(Perforce Path)

Changelist Num:
(one to grade)

Verified Build: Yes No
(Do not check unless you did verify)

YouTube Link:
(Please verify sound and quality)

Design Doc Path:
(Perforce Path to PDF)

Design Details

Design Patterns completed and use in your code:

Singleton	Factory	Observer
Flyweight	Proxy	Command
Iterator	State	Composite
Strategy	Visitor	Object Pools
Null Object	Adaptor	Template

Design Document PDF

Completed and submitted

Video Submitted

Demos each feature

Loud and Clear Sound (Can hear your voice easily)

Large Screen capture for Game Screen not the full desktop

Toggle Bounding Boxes on/off

Game Evaluation

Game Cycling

Select Screens --> Game Screens --> Game Over Screen --> Select Screens

Screens Completed

Select Screen

Game Screen

Game-Over Screen

Game Play

Complete at least 2 levels without crashing

Each level progressively harder and faster

Toggling bounding boxes on and off (Need to Demo)

2 Player mode

Player takes turns playing one at a time

Each player's game assets are unique to the player

Number of Aliens, Shield Damage, Score, etc.

Each player plays until they die, then next player plays if they still have lives left.

Once both players die, then go to Game Over screen

Score

High Score being updated

Different points for Aliens, UFOs, etc

Player

Movement

Number of lives

Launch on missile at a time

Collision Verification

Missile vs Shield

Missile vs Alien

Missile hits top of screen

Missile hits UFO

Missile hits Alien Bomb

Alien Grid versus each wall

Player Icon - right / left limits

Alien Bomb vs Player

Alien Bomb vs Floor

Alien Bomb vs Shield

Alien Grid

5x11 grid (Squid, Crab, Octopus)

Animation Movement in synchronization with movement

Progressively faster as number of aliens goes down

Music tempo and marching is faster as number of Aliens goes down

Drops different types of bombs (at least 3 different types) from bottom Alien

Explosion Splat when Alien is struck by missile

Explosion Splat when Alien Bomb is hit by missile

Killed Aliens adds points to player's score

Different points for different types of Aliens

UFO

Launches at different random intervals

Moves horizontally from Left or Right side

Drops Bomb

Plays unique Sound when UFO is on screen

Explosion sprite animation when UFO is killed

Killed UFO add points to Player's score

Missile / Bombs

Missile and Bombs can hit and kill each other

Splat animation when Missile and Bombs hit each other

Bombs Dropping (at least 3 different modes)

Zig Zag

Rolling

Straight

One Bomb at a time from each alien column (may have more)

Shields

4 Shields on the Screen

Impedes Missiles or Bombs from traveling through shield

Multiple hits can create holes in shields (need to demo this)

Graphical noise base Dissolve effects (optional)

Sounds

Audio of Aliens marching, synchronized with animation and movement

Cycles 4 sounds for the alien march sounds

Missile firing sound

Collision (missile hitting object)

Death sound effect

UFO

Sounds overlapping

Additional Material

Anything I should Know:

Verify Builds

- Follow the Piazza procedure on submission
 - Verify your submission compiles and works at the changelist number.
- Verify that only MINIMUM files are submitted
 - No – Generated files
 - *.pdb, *.suo, *.sdf, *.user, *.obj, *.exe, *.log, *.pdb, *.db, *.user
 - Anything that is generated by the compiler should not be included
 - No – Generated directories
 - /Debug, /Release, /Log, /ipch, /.vs
- Typical files project files that are required
 - *.sln, *.csproj, *.cs,
 - App.config, AssemblyInfo.cs, CleanMe.bat
 - Resources Directory:
 - *.tga, *.dll, *.wav, *.gls, *.azul

Standard Rules

Submit multiple times to Perforce

- Submit your work as you go to perforce several times (more than 20 submissions)
 - As soon as you get something working, submit to perforce
 - Have reasonable check-in comments
 - Points will be deducted if minimum is not reached

Submission Report

- Fill out the submission Report
 - No report, no grade

Code and project needs to compile and run

- Make sure that your program compiles and runs
 - Warning level 4
 - NO Warnings or ERRORS
 - Your code should be squeaky clean.
 - Code needs to work “as-is”.
 - No modifications to files or deleting files necessary to compile or run.
 - All your code must compile from perforce with no modifications.
 - Otherwise it's a 0, no exceptions

Project needs to run to completion

- If it crashes for any reason...
 - It will not be graded and you get a 0

No Containers

- Containers (No automatic containers or arrays)
- Template or generic parameters
- **No arrays**
 - You need to do this the old fashion way - **YOU EARNED IT**
 - Exception – **Glyphs** for fonts

Leave Project Settings

- Do NOT change the project or warning level
 - Any changing of level or suppression of warnings is an integrity issue

Simple C#

- No .Net
- We are using the basics
 - Types:
 - Class, Structs, intrinsic types (int, float, bool, etc...)
 - Basics language features
 - Inheritance, methods, abstract, virtual, etc...

No Debug code or files disabled

- Make sure the program has only active code
 - If you added debug code or commented out code,
 - please return to code to active state or remove it

Adding files to this project

- Make sure you add the files in the appropriate sub-directories
- Make sure any new files are successfully integrated into the project
- Make sure your new files are submitted to Perforce