

**Aventurile intergalactice ale robotilor**  
**Manea Paula**  
**Grupa 1211A**

**Povestea jocului:**

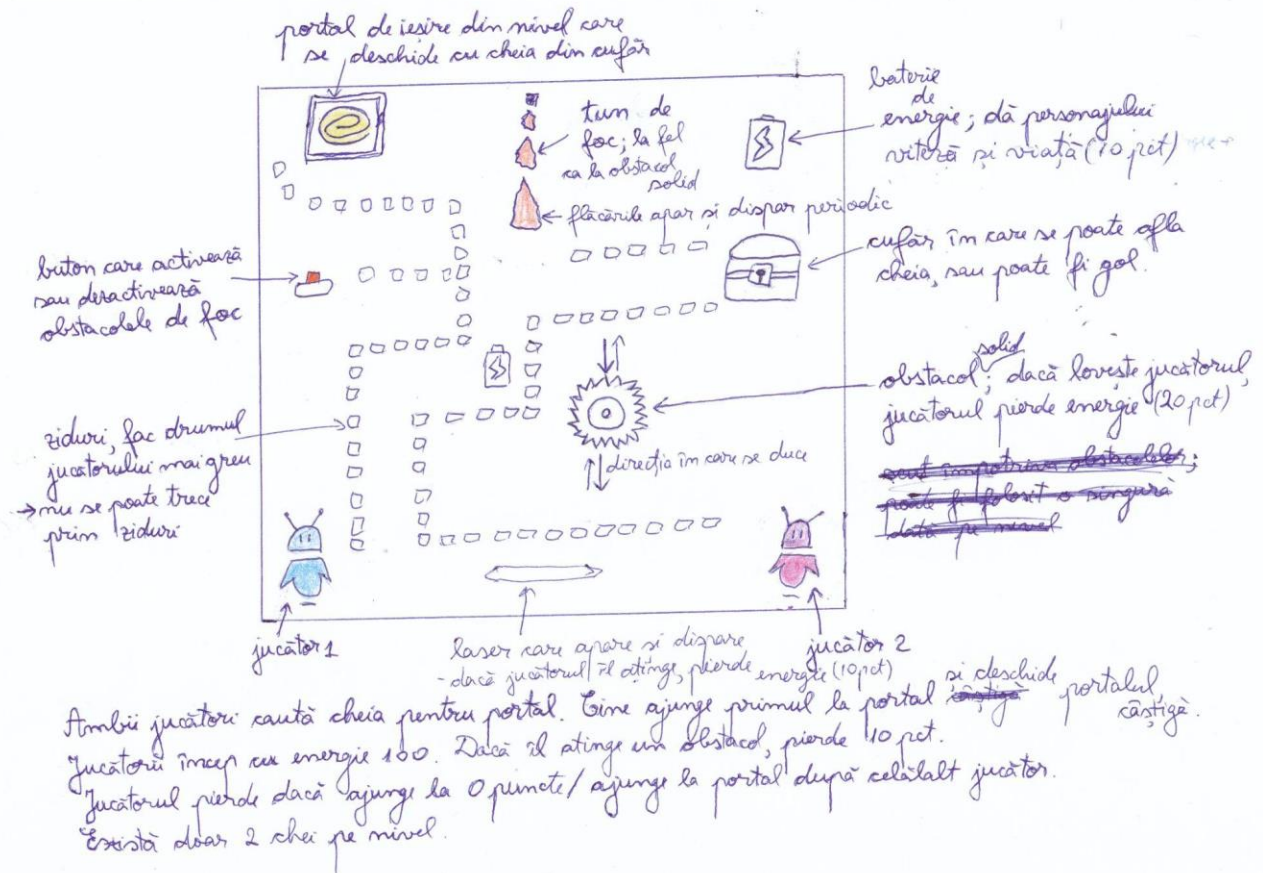
Ne aflam in galaxia Yuqo, in anul 12.000. Oamenii au părăsit de trei milenii planeta Pamant și s-au stabilit pe uimitoarea planetă gigant "Planet X-347", unde au resursele necesare vieții.

Într-o zi de vară, dorind sa schimbe viata lor monotona intr-una palpitanta, prietenii Andy si Mark decid sa construiasca si sa isi programeze proprii roboti, pe care ii numesc Kevin si Erika. Cei doi amici își trimit robotii intr-o calatorie intergalactica pentru a ajunge pe Pamant. Aceasta calatorie nu se poate face foarte rapid, căci portalul direct către vechea planeta a oamenilor este stricat. Robotii Kevin și Erika trebuie astfel să treacă prin trei planete diferite pentru a ajunge la destinatie: planetele Nobu, Kora si Minty.

Drumul indirect către Pamant vine însă cu provocări. Guvernatorii celor trei planete decreta urmatorul lucru: Robotilor nu li se da voie sa treacă prin cele trei planete decat daca gasesc cheile de activare a portalurilor si evita obstacolele existente pe planetele respective.

Andy si Mark accepta provocarea. Mai mult, tinerii sunt foarte competitivi si isi propun sa vada care dintre roboti ajunge primul pe Pamant. Controlati de la distanta, robotii trebuie sa caute surse de energie și sa combata obstacole periculoase în încercarea de a deveni castigatori. Oare cine va invinge?

## Proiectarea contextului:



## Proiectarea sistemului:

## Controls

Idle - automat ; nu este nevoie de player input



Mers stanga - buton A pentru jucător 1,  
buton stanga pentru jucător 2



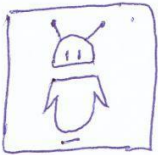
Mers dreapta - buton D pentru jucător 1,  
tastă dreapta pentru jucător 2



Mers sus - buton X pentru jucător 1,  
tastă sus pentru jucător 2

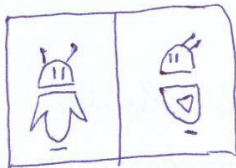


Mers jos - tasta S pentru jucător 1,  
săgeată jos pentru jucător 2



Camera - automată

- modul de bază al camerei este front view
- camera urmărește jucătorul
- sunt două camere, una pentru fiecare jucător



**\*este o singura camera, centrata pe un jucator**

**Atac – tasta C pentru jucatorul 1, tasta P pentru jucatorul 2**



**Colectarea bateriilor/ cheii se face automat cand se ajunge in zona lor. Activarea/ dezactivarea obstacolelor se face automat cand personajul ajunge pe buton.**

## Proiectarea conținutului:

Proiectarea conținutului

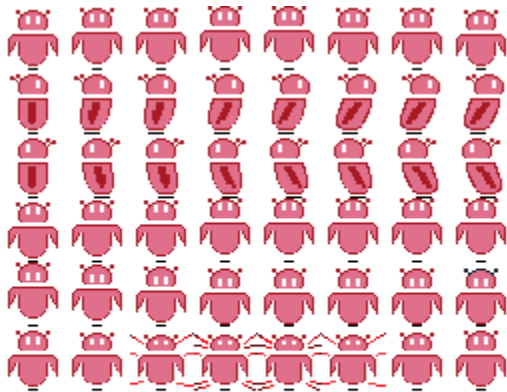
Personaje: 2 roboți

Robotul Kevin

Robotul Erika



- robotul Kevin



- robotul Erika

Titlu joc: Aventurile intergalactice ale robotilor

Descriere joc: Un joc de strategie interactiv, care pune la incercare agilitatea si inventivitatea jucatorilor.

Durata jocului: aprox. 15 minute

Audienta target: tineri

Pentru fiecare nivel exista componente pasive, cat si componente active.

Componente pasive: cutarul in care exista cheia ce deschide portalul, bateriile de energie care dau viata personajului, pereti care delimiteaza drumul, decorul specific fiecarei planete

Componente active: laserul, tunul de foc, butonul de activare/deactivare al tunului, obstacol solid care se misca in sus si in jos

Obiectiv: ghidarea robotilor Kevin si Erika catre portalul de iesire din nivel, colectand cheia de deschidere a portalului si incercand sa ramana cu cat mai multa viata.

Provocari: Cu cat avanseaza personajele in joc, cu atat nivelurile devin mai solicitante, avand obstacole noi.

**Componente:**

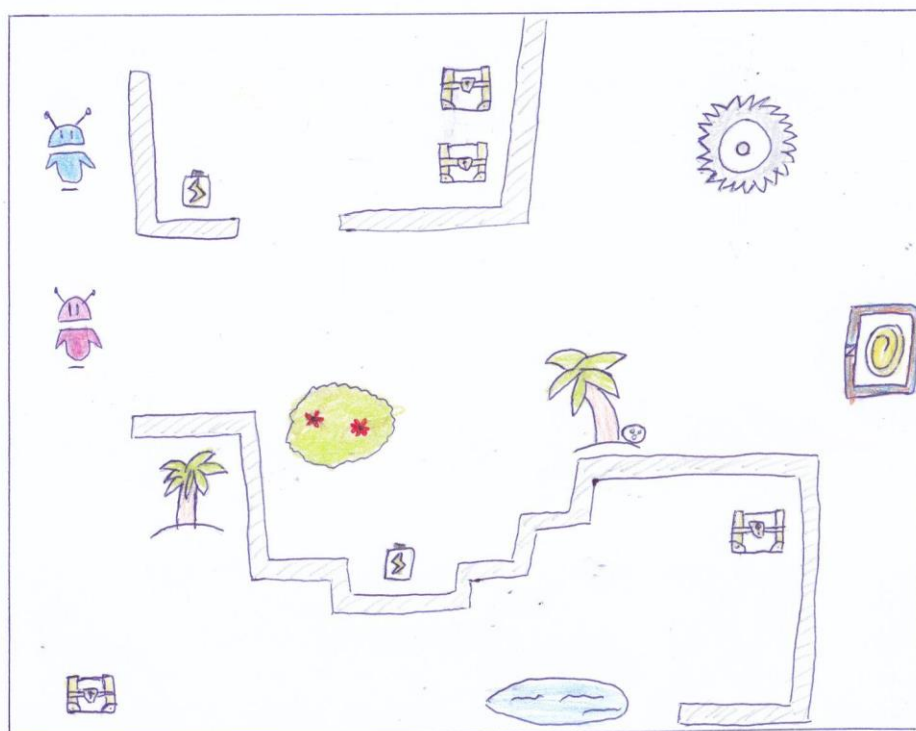
Cufar, baterie, portal, buton de activare/dezactivare obstacole, respectiv cele trei obstacole.

**Reguli joc:**

Jocul este unul 1v1. Ambii jucători se plimba pe harta și încearca sa ramana in viata cat mai mult timp posibil, evitand obstacolele care le stau in cale. Castigatorul este cel care găsește cheia de deschidere a portalului către nivelul următor. Daca amandoi găsesc cheia si ajung in același timp la portal, se decide ca amandoi sa fie numiți castigatori iar departajarea se face la următorul nivel.

**Proiectarea nivelurilor:**

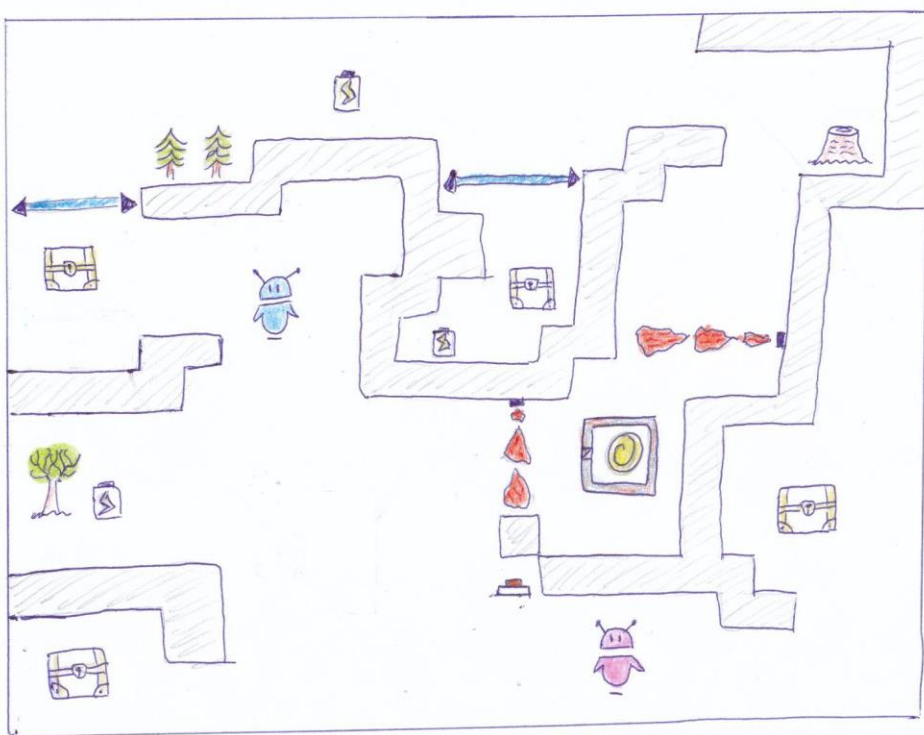
Sunt trei niveluri. În fiecare nivel jucătorii ajung pe o planetă diferită: Planeta Nisipului, Planeta Pădurilor și Planeta Diamantelor. La fiecare nivel diferă numărul obstacolelor, al cuferelor și al bateriilor de energie care se găsesc in calea celor doi roboți. Trecerea de toate cele trei niveluri aduce cu sine sfarsitul jocului si atingerea obiectivului inițial: planeta Pamant.



N  
i  
v  
e  
l  
  
1

Sand land

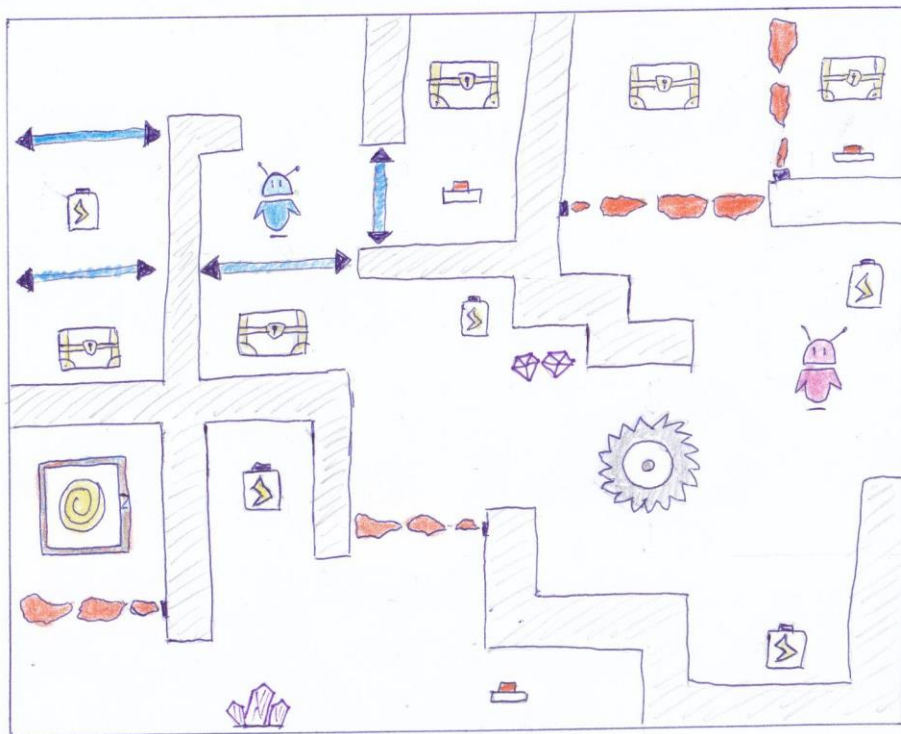




N  
i  
V  
E  
L  
2

Forest land

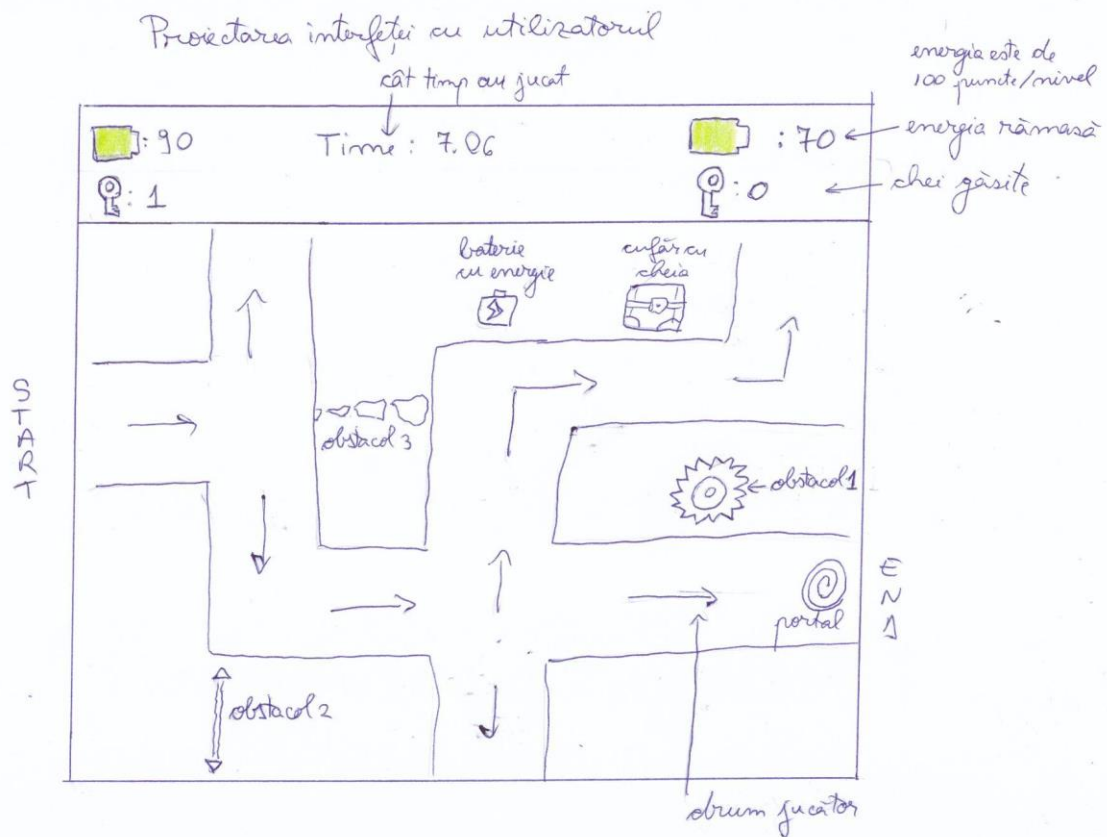




N  
i  
v  
e  
l  
3

Gem land

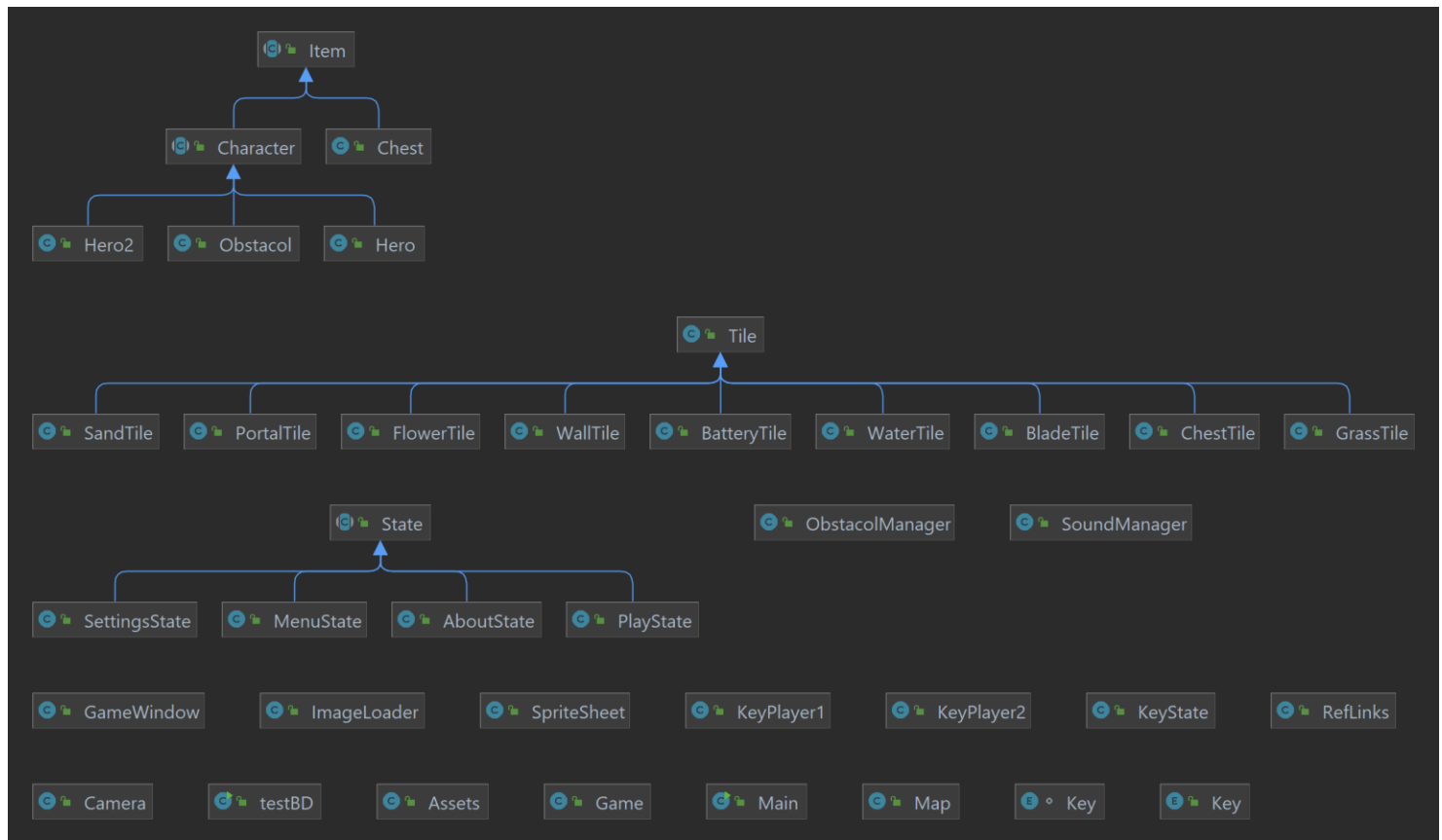
## Proiectarea interfeței cu utilizatorul:



## Interfața cu obiectele din joc:

- Jucătorul interacționează cu butoanele de activare/dezactivare a tunurilor de foc. Atunci când apasă pe buton, unul sau mai multe din tunurile plasate la distanța care emana foc se oprește.
- Jucătorul interacționează și cu laserele. Laserele emit energie periodic. Atunci când e inactiv, jucătorul poate trece prin zona laserului.
- Cufarul este alt element cu care poate interacționa. În prezența jucătorului, cufarul se deschide, iar dacă există în cufar o cheie, cheia este contorizată și intra în posesia jucătorului.
- Interacțiunea cu portalul se poate face abia dacă jucătorul deține cheia de deschidere a acestuia. Fără cheie, portalul rămâne blocat.
- Nu se poate interacționa cu elementele de decor specifice terenului fiecărui nivel.

## Diagrama UML :



Item		
m	Item(RefLinks, float, float, int, int)	
m	GetHeight ()	int
m	SetX(float)	void
m	Draw(Graphics)	void
m	SetHeight(int)	void
m	GetY()	float
m	SetWidth(int)	void
m	intersectHeros (Rectangle, Rectangle)	boolean
m	Update ()	void
m	GetWidth()	int
m	GetX()	float
m	SetY(float)	void
m	SetAttackMode ()	void
m	SetNormalMode ()	void

Character		
m	Character(RefLinks, float, float, int, int)	
m	SetSpeed (float)	void
m	GetSpeed ()	float
m	SetLife(int)	void
m	SetScore (int)	void
m	Move()	void
m	GetLife ()	int
m	GetScore ()	int
m	GetYMove()	float
m	MoveX()	void
m	MoveY()	void
m	GetXMove ()	float
m	SetXMove (float)	void
m	SetYMove (float)	void

Obstacol		
m	Obstacol(RefLinks, float, float, int, int, float, float)	
m	Update ()	void
m	getBounds ()	Rectangle
m	Draw(Graphics)	void

Hero		
m	Hero(RefLinks, float, float)	
m	GetInput()	void
m	Update ()	void
m	intersectareSpike ()	void
m	Draw(Graphics)	void
m	getBounds ()	Rectangle

Hero2		
m	Hero2(RefLinks, float, float)	
m	getBounds ()	Rectangle
m	GetInput()	void
m	Draw(Graphics)	void
m	Update ()	void

Map		
m	Map(RefLinks)	
m	Update ()	void
m	Draw(Graphics)	void
m	findBladeY ()	int
m	GetTile(int, int)	Tile
m	getHeight ()	int
m	checkCollisionWithObstacles (float, float, int)	boolean
m	checkCollisionWithBattery (float, float, int)	boolean
m	getWidth ()	int
m	MiddleEastMap (int, int)	int
m	LoadWorld ()	void

RefLinks		
m	RefLinks(Game)	
m	SetGame (Game)	void
m	GetMap ()	Map
m	SetMap (Map)	void
m	GetKeyManagerState ()	KeyState
m	GetWidth ()	int
m	GetHeight ()	int
m	GetKeyManager2 ()	KeyPlayer2
m	GetGame ()	Game
m	GetKeyManager ()	KeyPlayer1

Camera		
m	Camera(int, int)	
m	getY()	int
m	centerOnEntity(Hero)	void
m	setX(int)	void
m	getPastY()	int
m	Draw(Graphics)	void
m	getX ()	int
m	getPastX ()	int
m	setY(int)	void

ObstacolManager		
m	ObstacolManager ()	
m	removeObstacle (Obstacol)	void
m	drawObstacles (Graphics)	void
m	updateObstacles ()	void
m	getInstance ()	ObstacolManager
m	addObstacle (Obstacol)	void

GameWindow		
m	GameWindow (String, int, int)	
m	GetWndWidth ()	int
m	GetWndFrame ()	JFrame
m	GetCanvas ()	Canvas
m	BuildGameWindow ()	void
m	GetWndHeight ()	int

Key		
m	Key(int)	
m	getInfo ()	int
m	valueOf(String)	Key
m	values ()	Key[]

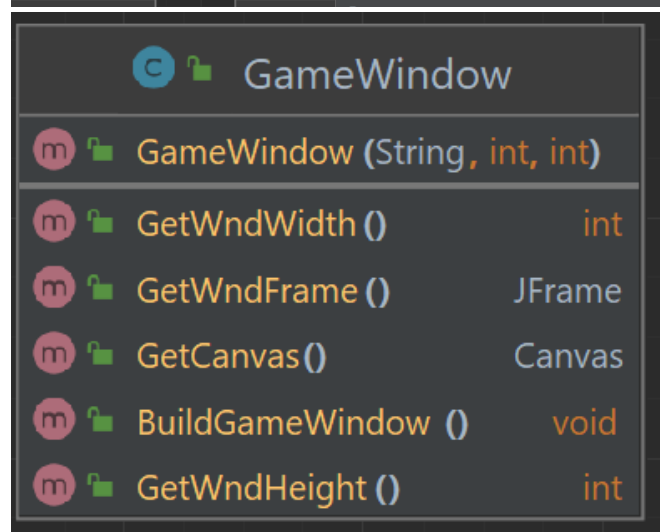
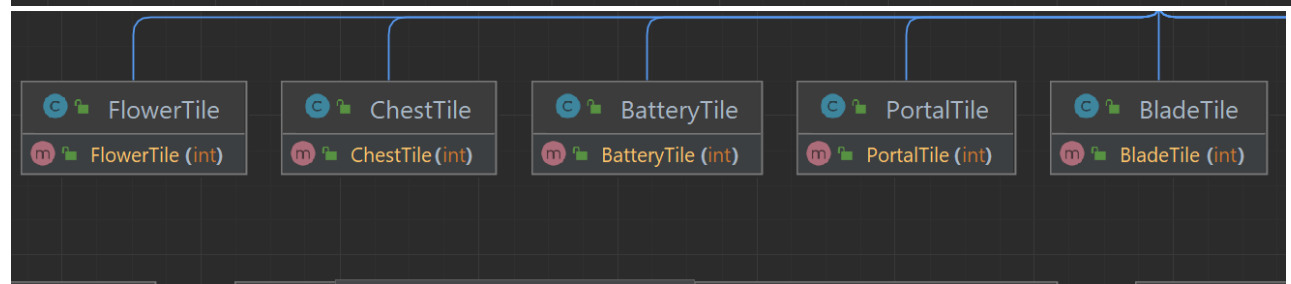
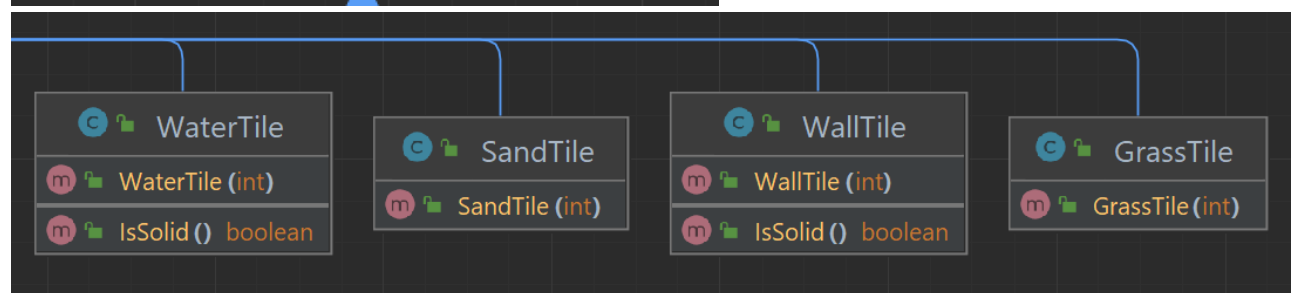
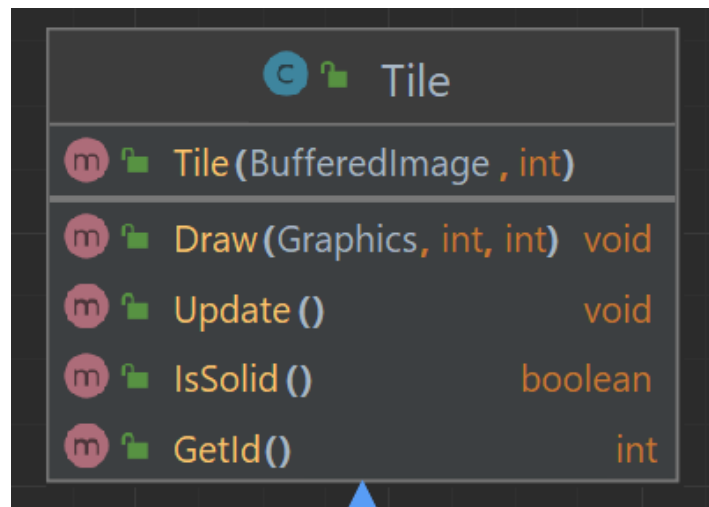
ImageLoader		
m	ImageLoader ()	
m	LoadImage (String)	BufferedImage ?

SoundManager		
m	SoundManager ()	
m	playSound (String)	void

testBD		
m	testBD ()	
m	main (String[])	void

Main		
m	Main ()	
m	main (String[])	void

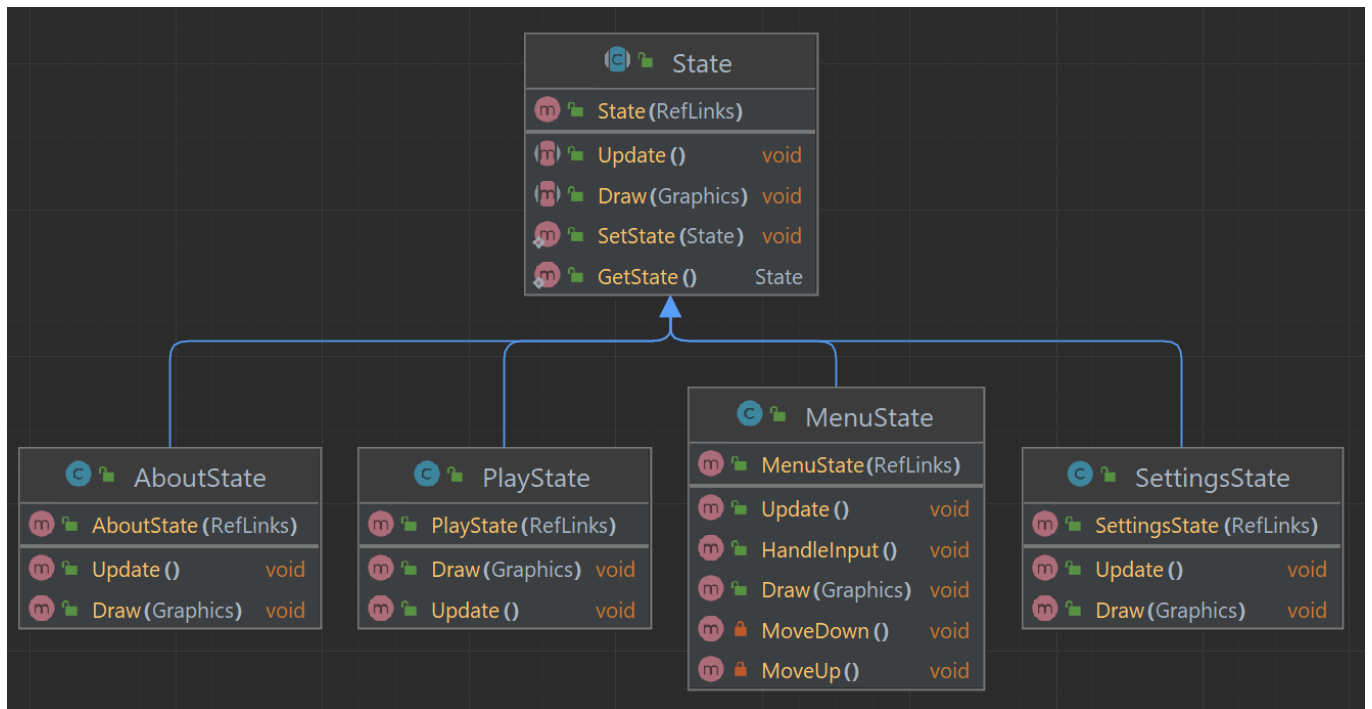
Assets		
m	Assets ()	
m	Init ()	void





KeyPlayer1	KeyState	KeyPlayer2
KeyPlayer1 ()	KeyState ()	KeyPlayer2 ()
Update () void	keyTyped (KeyEvent) void	keyTyped (KeyEvent) void
keyTyped (KeyEvent) void	Update () void	Update () void
keyPressed (KeyEvent) void	keyReleased (KeyEvent) void	keyPressed (KeyEvent) void
keyReleased (KeyEvent) void	keyPressed (KeyEvent) void	keyReleased (KeyEvent) void

Game
Game (String, int, int)
GetHeight () int
StartGame () void
run() void
SetState (State) void
GetWidth () int
GetKeyManager2 () KeyPlayer2
GetKeyManager3 () KeyState
StopGame () void
GetKeyManager () KeyPlayer1
InitGame () void
Draw () void
Update () void



## Bibliografie

<https://www.gamedeveloper.com/disciplines/game-descriptions-rules-and-mechanics-what-are-the-differences-and-similarities->

["Rulebook" vs "How to Play" - Learning & Teaching Games](#)

[Pyxel Edit: pixel art and tileset creation tool](#)